






SPIS TREŚCI NUMERU 10 (425)

O topologii programów współbieżnych <i>Maria Donten</i>	str. 1
Kosmiczna linijka 10. Wielki Atraktor: dokąd płyną galaktyki? <i>Bożena Czerny, Agnieszka Janiuk</i>	str. 5
Konkurs zadań astronomicznych	str. 5
Kamery CCD w astronomii <i>Wojciech Pych</i>	str. 6
 Zadania	str. 7
Pamięć cache w praktyce <i>Krzysztof Piecuch</i>	str. 8
Co to znaczy zrekonstruować porządek? <i>Marta Przyborowska</i>	str.10
 Czy komputer potrafi rysować wykresy funkcji? <i>Piotr Krzyżanowski</i>	str.12
Czy naprawdę <i>prawie</i> robi wielką różnicę? <i>Paulina Malolepsza, Tomasz Malolepszy</i>	str.14
Informatyczny kącik olimpijski (24): Barykady <i>Tomasz Kulczyński</i>	str.16
O własności Darboux <i>Witold Bednarek</i>	str.17
 Wyznaczamy moduł Younga <i>Stanisław Bednarek</i>	str.18
Wolniejszy wyprzedza <i>Radosław Poleski</i>	str.19
Aktualności	str.20
 Stowarzyszenie na rzecz Edukacji Matematycznej	str.21
Klub 44	str.22
Patrz w niebo	str.24
Październik	str.24
 Gry <i>Joanna Jaszewska</i>	str.25

Miesięcznik *Delta* – matematyka, fizyka, astronomia, informatyka jest wydawany przez Uniwersytet Warszawski przy współpracy towarzystw naukowych: Polskiego Towarzystwa Matematycznego, Polskiego Towarzystwa Fizycznego, Polskiego Towarzystwa Astronomicznego i Polskiego Towarzystwa Informatycznego.

Komitet Redakcyjny: dr Piotr Chrzastowski-Wachtel, dr Krzysztof Ciesielski – wiceprzewodniczący, prof. dr hab. Bożena Czerny, dr Andrzej Dąbrowski, prof. dr hab. Krzysztof Diks, prof. dr hab. Jan A. Gaj – przewodniczący, prof. dr hab. Jerzy Ginter, dr Piotr Goldstein, dr Zofia Gołąb-Meyer, prof. dr hab. Paweł Idziak, dr Agnieszka Janiuk, dr Marcin Kiraga, dr hab. Andrzej Majhofer, dr hab. Zbigniew Marciniak, dr hab. Zygmunt Mazur, dr Adam Michalec, dr Zdzisław Pogoda, prof. dr hab. Wojciech Rytter, prof. dr hab. Grzegorz Sitarski, dr hab. Paweł Strzelecki.

Redaguje kolegium w składzie: Marcin Adamski, Wiktor Bartol, Ewa Czuchry, Maria Donten, Krystyna Kordos – sekr. red., Marek Kordos – red. nac., Tomasz Kwast, Agnieszka Majczyna, Jakub Radoszewski, Anna Rudnik, Witold Sadowski, Krzysztof Turzyński, Andrzej Walat, Piotr Zalewski – z-ca red. nac.
Okładki i ilustracje: Emilia Bojańczyk.

Adres do korespondencji:
Instytut Matematyki UW, Redakcja „Delta”, ul. Banacha 2, pokój 4020, 02-097 Warszawa, e-mail: delta@mimuw.edu.pl, tel. 022-55-44-402.

Skład systemem T_EX oraz rysunki techniczne wykonała Redakcja.
Wydrukowano w Drukarni Greg, ul. Konstruktorska 4, 02-673 Warszawa.

WARUNKI PRENUMERATY W FIRMIE AMOS

01-785 Warszawa, ul. Broniewskiego 8A (tel. 022-639-73-67)
internet: www.amos.waw.pl , e-mail: biuro@amos.waw.pl
Wpłaty przyjmowane są non-stop, do 10. dnia miesiąca poprzedzającego okres prenumeraty. **Okres prenumeraty wynosi co najmniej trzy miesiące.** Cena jednego numeru w 2010 roku wynosi 4 zł. Przy wpłacie prosimy o zaznaczenie okresu prenumeraty.

W prenumeracie zagranicznej (też przez okres **co najmniej trzech miesięcy**) cena numeru w 2010 r. wynosi 8 zł. W przypadku życzenia dostawy priorytetowej odpowiednią dopłatę ponosi zamawiający.

Uwaga! Dla zamawiających minimum 10 egzemplarzy każdego numeru AMOS funduje dodatkowo jeden egzemplarz pisma.

Konto AMOS-u: PKO BP SA I O/W-wa, nr 11 1020 1013 0000 0502 0004 0584

WARUNKI PRENUMERATY W RUCH-u

internet www.ruch.com.pl, infolinia 0-804-200-600

Cena prenumeraty w 2010 roku wynosi 4 zł za egzemplarz.

1. **Prenumerata krajowa:** wpłaty przyjmują jednostki kolportażowe „RUCH” SA właściwe dla miejsca zamieszkania. Termin przyjmowania prenumeraty: do 5. każdego miesiąca poprzedzającego okres rozpoczęcia prenumeraty.

2. **Prenumerata ze zleceniem wysyłki za granicę:** informacji o warunkach prenumeraty i sposobie zamawiania udziela „RUCH” SA Oddział Krajowej Dystrybucji Prasy, 01-248 Warszawa, ul. Jana Kazimierza 31/33; tel. 022-5328-731 (prenumerata płatna w walucie obcej), -816, -734, -819 (prenumerata płatna w PLN w kasie Oddziału lub na konto w banku PEKAO SA IV O/Warszawa 68 1240 1053 1111 0000 0443 0494), infolinia 0-800-1200-29.

3. **Prenumerata opłacana za granicą:** przelewem na nasze konto SWIFT banku: PKOPPLPWWA4; w USD PEKAO SA IV O/W-wa IBAN PL54 1240 1053 1787 0000 0443 0508; w EUR PEKAO SA IV O/W-wa IBAN PL54 1240 1053 1978 0000 0443 0511; kserokopię polecenia przelewu należy przesłać faksem pod numer +48-22-5328-731.

Numerzy archiwalne (od 1987 r.) można nabyć w Redakcji osobiście lub listownie.

Strona internetowa (streszczenia, artykuły archiwalne, linki itd.):
<http://www.mimuw.edu.pl/delta>

W następnym numerze:
Grawitino

Wydawca: Uniwersytet Warszawski

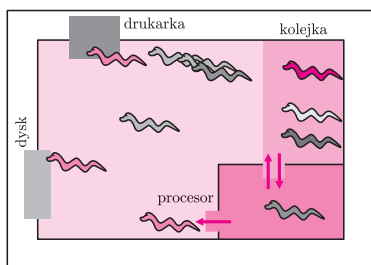
Cena 1 egzemplarza 4 zł

Czy informatycy zajmują się dowodzeniem twierdzeń? Zdarza się, że poprawność programu, czy pomysłu na program, jest nieoczywista. Wtedy potrzebne są metody, które pozwolą przeprowadzić ścisły dowód, że zaprojektowany algorytm zadziała tak, jak powinien. W niektórych zagadnieniach pomysły na dowód sięgają w głąb matematyki.



Przyjrzyjmy się paradygmatowi programowania współbieżnego. Jeśli chcemy rozwiązać duży problem obliczeniowy, to dobrym pomysłem jest podzielenie go na podproblemy. Dzięki temu, jeśli w obliczeniach pojawi się błąd, wystarczy powtórzyć jedynie część, która nie została wykonana poprawnie. Jeśli mamy odpowiedni sprzęt, np. kilka komputerów lub komputer z kilkoma procesorami, możemy przyspieszyć obliczenia, pracując nad wieloma podproblemami w tym samym czasie. W ten sposób podejście sekwencyjne zastępujemy przez współbieżne. Opisana idea współbieżności jest bardzo ogólna – można ją stosować nie tylko do obliczeń komputerowych, lecz także do rozwiązywania dowolnych problemów. Szczegółami zajmiemy się jednak tylko w przypadku programowania.

Podstawowym pojęciem tej dziedziny jest proces sekwencyjny (zwykle nazywany po prostu procesem), czyli realizacja programu w pewnym środowisku lub wstrzymana realizacja w oczekiwaniu na pewne zdarzenie. Realizacja programu to ciąg wykonania kolejnych instrukcji. Często wykonanie jakiejś operacji wymaga sprzyjających warunków. Jeśli trzeba na nie poczekać, wykonanie programu jest na chwilę wstrzymywane. Proces to byt bardzo abstrakcyjny – czy „ciąg wykonania instrukcji” można sobie jakoś wyobrazić, narysować? Otóż można: wystarczy do każdego ciągu instrukcji dodać stworzonko, które odpowiada za jego wykonanie, czyli wędruje po wnętrzu komputera, realizując kolejne zadania w odpowiednich miejscach. Na przykład stoi w kolejce do drukarki, zapisuje coś na dysku, czeka na przydział czasu do obliczeń na procesorze lub właśnie wykonuje obliczenia (rys. 1). Ten obrazek pozwala mniej więcej wyobrazić sobie, co się odbywa w systemie operacyjnym podczas wykonywania programów. Jego wersje z nieco większą liczbą szczegółów przydają się wszystkim studentom, którzy zaliczają laboratorium z programowania współbieżnego. . .



Rys. 1. Życie codzienne procesów.

Dążymy do tego, żeby rozwiązywać problemy programistyczne, rozdzielając pracę między kilka procesów działających w tym samym czasie. Dokładniej, dwa procesy nazwiemy współbieżnymi, jeśli jeden z nich rozpoczął działanie po rozpoczęciu, ale przed zakończeniem działania drugiego. Program współbieżny składa się z kilku procesów współbieżnych, wykonujących elementy pewnego zadania. I tu pojawiają się trudności. . . Jeśli procesy pracują razem nad jednym problemem, korzystają ze wspólnych danych, potrzebują nawzajem swoich częściowych wyników, to muszą się komunikować, żeby nie doszło do żadnych nadużyć.

Najprostszy przykład możliwości nadużycia jest problem sekcji krytycznej, czyli ustalenia, który proces uzyska dostęp do zasobu, z którego w danej chwili może korzystać tylko jeden proces. Przyjrzyjmy się prostemu programowi, w którym działają dwa procesy, P i Q . Każdy z nich wykonuje następujący ciąg czynności: wczytuje wartość zmiennej globalnej v na swoją zmienną lokalną, zwiększa jej wartość o 1 i zapisuje wynik na zmiennej v .

```

var v: integer; v := 0;
process P;
var P.i, P.x: integer;
P.x := v;
P.x := P.x + 1;
v := P.x;
end;
process Q;
var Q.i, Q.x: integer;
Q.x := v;
Q.x := Q.x + 1;
v := Q.x;
end;
    
```

Zmienna v ma początkowo wartość 0, każdy proces zwiększa jej wartość o 1, więc po zakończeniu działania programu v powinna mieć wartość 2. Czy na pewno ten program tak właśnie zadziała? Często tak, na przykład w ekstremalnym przypadku, kiedy proces P rozpocznie działanie tuż przed wykonaniem końcowej operacji procesu Q , ale nie zawsze. Sytuacja komplikuje się, jeśli procesy wykonują te same instrukcje w prawie tym samym czasie. Powiedzmy, że P wczytuje wartość zmiennej v i, zanim zdąży wykonać kolejną czynność, Q rozpoczyna działanie i również wczytuje wartość v – zmienne lokalne $P.x$ i $Q.x$ mają wartość 0. Jeśli

Jak procesy się komunikują? Mają swoje sposoby, zależne od języka programowania lub abstrakcyjnej notacji używanej do opisu programów współbieżnych. Na przykład, nadają komunikat do konkretnego adresata, który musi w odpowiedniej chwili nastawić się na odbiór (CSP), albo umawiają się na randki (Ada), albo wrzucają dane do worka i wyciągają je według szablonów (Linda). . .

Oczywiście, procesy danego programu nie muszą wykonywać tego samego (lub symetrycznego) kodu, tak jak w przykładzie.

*doktorantka, Instytut Matematyki UW

Okazuje się, że ominięcie tego problemu wcale nie jest łatwe. . . Właściwie jest niemożliwe bez dodatkowych mechanizmów służących do synchronizacji procesów, zdefiniowanych na poziomie systemu operacyjnego.

W literaturze informatycznej w języku polskim często pozostawiane są angielskie nazwy problemów i algorytmów, co ułatwia porównanie z literaturą anglojęzyczną. Również tutaj stosujemy tę konwencję.



teraz P zwiększy wartość $P.x$ i zapisze ją na v , a potem Q wykona te same instrukcje, to zmienna v będzie miała wartość 1, chociaż każdy z procesów wykonał dodawanie! Wynik jest niezgodny z oczekiwaniami. Powstał błąd, ponieważ procesy nie uzgodniły, który najpierw zmodyfikuje globalną. Proces modyfikacji składa się z kilku instrukcji: wczytania, zwiększenia i zapisania. Język programowania nie zapewnia, że te wszystkie czynności zostaną wykonane w jednym, nieprzerwanym bloku – zadbanie o to jest zadaniem programisty!

Nawet z tego prostego przykładu widać, że sprawdzenie, czy program współbieżny jest poprawny, może być bardzo trudne. Trzeba sprawdzić, czy działanie programu jest zgodne z oczekiwaniami dla każdego przeplotu, czyli dla każdej sekwencji wykonań instrukcji procesów biorących udział w programie – począwszy od przypadków, w których czasy wykonania procesów prawie się nie pokrywają, aż do takich, w których procesy wykonują na zmianę bardzo małe fragmenty kodu. Takich przeplotów może być bardzo dużo, w rzeczywistych problemach często nieskończenie wiele. . . A oprócz bezpieczeństwa programu, czyli zapewnienia, że w każdym momencie w sekcji krytycznej będzie co najwyżej jeden proces, należy zadbać jeszcze o inne własności. Między innymi nie można zagłodzić procesu, czyli spowodować, że nie będzie mógł nigdy wykonać pewnej instrukcji swojego kodu, bo, na przykład, inne procesy będą się zawsze przed nim wpychać do kolejki. Nie można również dopuścić do zakleszczenia – sytuacji, w której grupa procesów oczekuje na zdarzenie, które może spowodować tylko proces z tej grupy.

Pytanie, czy dla danego problemu istnieje poprawny program współbieżny, który go rozwiązuje, wygląda na jeszcze trudniejsze niż sprawdzenie poprawności napisanego kodu. Jednak dość niedawno powstało matematyczne kryterium, które pozwala spojrzeć na to pytanie z zupełnie innej strony, a często również udzielić kompletnej odpowiedzi. Oczywiście nie dla zupełnie dowolnych problemów. Interesująca nas klasa to problemy (lub zadania) decyzyjne (ang. *decision tasks*), w których współbieżność jest w pewien sposób wbudowana w sformułowanie zadania. Inaczej mówiąc, takie, których istotą jest dokonanie pewnych uzgodnień pomiędzy procesami. Procesy uczestniczące w zadaniu dostają dane wejściowe, zazwyczaj liczby, wykonują pewien algorytm i podają wynik, który musi być zgodny ze specyfikacją problemu. Zobaczmy kilka przykładów.

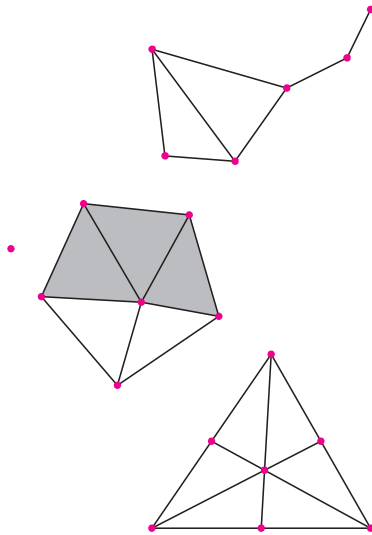
Najprostszy w sformułowaniu (ale nie w rozwiązaniu) problem to *Consensus*. Uruchamiamy pewną liczbę procesów, każdy z nich dostaje wartość wejściową 0 lub 1. Jako wynik wszystkie procesy mają podać tę samą wartość x , będącą jedną z wartości, które dostały na wejściu. W szczególności, jeśli wartości wejściowe były równe, to jest tylko jeden dopuszczalny wynik; w pozostałych przypadkach procesy mają większą dowolność w wyborze wyniku. Może się zdarzyć, że nie wszystkie procesy, które rozpoczną działanie, zakończą się poprawnie i podadzą wynik. Wtedy chcemy wymagać, żeby procesy, w których nie wystąpił błąd, zachowywały się zgodnie z założeniami zadania – wynik ma być zgodny ze specyfikacją dla przypadku, w którym uruchomione zostałyby tylko te procesy, które zadziałały bezbłędnie.

Dalej przyda się nam problem *Binary Consensus*, będący ograniczeniem powyższego do przypadku dwóch procesów, i jego modyfikacja, *Quasi-Consensus*. Zmiana polega na dopuszczeniu jeszcze jednego wyniku: jeśli dane wejściowe procesów P i Q są różne, to oprócz odpowiedzi $P \mapsto 0, Q \mapsto 0$ i $P \mapsto 1, Q \mapsto 1$ za prawidłową uznajemy również $P \mapsto 1, Q \mapsto 0$ (ale odwrotnej już nie!).

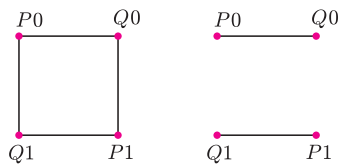
Bardziej złożonym zadaniem jest *Unique-Id* – w systemie działa n procesów, wartości wejściowe są nieistotne (np. wszystkie są zerami), procesy muszą wybrać różne wartości ze zbioru $\{0, \dots, n-1\}$. Ciekawy przykład „z życia” to *Renaming* – procesy dostają różne wartości z dużego zbioru $\{0, \dots, M\}$, zwracają różne wartości z mniejszego zbioru $\{0, \dots, K\}$. To zadanie można przetłumaczyć na problem układania bezkolizyjnych tras dla samolotów. Wystarczy podawać numery identyfikacyjne lotów na wejściu, a wyniki potraktować jako numery korytarzy powietrznych, których zazwyczaj jest istotnie mniej niż wszystkich samolotów.

Jak wygląda precyzyjny matematyczny opis interesujących nas zadań? Zestawy wartości wejściowych i wyjściowych będziemy zapisywać jako wektory – mogą w nich występować liczby lub wartość nieokreślona, oznaczana \perp (nie dopuszczamy wektora złożonego tylko z wartości nieokreślonych; to kwestia notacji). W wektorze

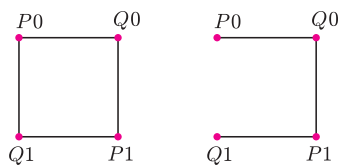
Warto spróbować zinterpretować te założenia, pamiętając, jakie znaczenie ma \perp w wektorach zbiorów \mathcal{I} i \mathcal{O} .



Rys. 2. Kilka kompleksów symplecjajnych.



Rys. 3. Kompleksy \mathcal{I} i \mathcal{O} dla zadania *Binary Consensus*.



Rys. 4. Kompleksy \mathcal{I} i \mathcal{O} dla zadania *Quasi-Consensus*.

Jak wyrazić założenia, które przyjęliśmy dla \mathcal{I} , \mathcal{O} i Δ , w języku odpowiadających im kompleksów i przekształcenia?

wejściowym wartość \perp na i -tym miejscu oznacza, że i -ty proces nie bierze udziału w przebiegu programu, natomiast w wektorze wyjściowym informuje, że i -ty proces nie zakończył działania poprawnie – w ogóle nie brał udziału w danym przebiegu lub zakończył się z błędem. Zbiory wektorów wejściowych i wyjściowych oznaczamy odpowiednio \mathcal{I} i \mathcal{O} . Potrzebna jest nam jeszcze relacja $\Delta \subseteq \mathcal{I} \times \mathcal{O}$ między tymi zbiorami, określająca, które wyniki uznajemy za poprawne dla różnych wektorów wejściowych. Zwykle będziemy patrzeć na Δ jako na funkcję z \mathcal{I} w zbiór wszystkich podzbiorów \mathcal{O} . To już prawie wystarczy – trzeba jeszcze założyć, że \mathcal{I} , \mathcal{O} i Δ mają pewne dobre własności. Zbiory wektorów mają być zamknięte na prefiksy: dla każdego wektora $\alpha \in \mathcal{I}$ w zbiorze \mathcal{I} muszą być również wszystkie wektory, które można otrzymać z α , zastępując niektóre liczby przez \perp ; to samo dla \mathcal{O} . Natomiast od Δ wymagamy, żeby w obrazie wektora α były tylko wektory, które mają \perp na tych samych pozycjach co α .

Trójka $(\mathcal{I}, \mathcal{O}, \Delta)$, której elementy spełniają powyższe założenia, to kompletny opis zadania. Problem *Binary Consensus* możemy więc sformułować, definiując

$$\mathcal{I} = \{(0, \perp), (1, \perp), (\perp, 0), (\perp, 1), (0, 0), (0, 1), (1, 0), (1, 1)\},$$

$$\mathcal{O} = \{(0, \perp), (1, \perp), (\perp, 0), (\perp, 1), (0, 0), (1, 1)\},$$

oraz $\Delta(\alpha) = \{\alpha\}$, jeśli α zawiera \perp , a w pozostałych przypadkach

$$\Delta((0, 0)) = \{(0, 0)\}, \quad \Delta((1, 1)) = \{(1, 1)\},$$

$$\Delta((1, 0)) = \Delta((0, 1)) = \{(0, 0), (1, 1)\}.$$

Opis zadania *Quasi-Consensus* otrzymujemy z powyższego, zmieniając tylko wartość Δ dla wektorów $(0, 1)$ i $(1, 0)$:

$$\Delta((1, 0)) = \Delta((0, 1)) = \{(0, 0), (1, 1), (1, 0)\}.$$

Uzyskałliśmy prosty, kombinatoryczny sposób opisu problemów, które chcemy rozwiązywać. Jednak dla trudnych zadań, w których uczestniczy wiele procesów, a takie zwykle pojawiają się w życiu, struktury kombinatoryczne są duże, bardzo złożone, trudno coś na ich temat udowodnić. . . A gdyby spojrzeć na te obiekty „z daleka”, spróbować dostrzec ich kształt, jakieś charakterystyczne cechy, które pozwolą je lepiej zrozumieć bez analizowania wszystkich szczegółów? I tu wreszcie wkracza topologia!

Okazuje się, że zbiory \mathcal{I} i \mathcal{O} opisują kompleksy symplecjajne – przestrzenie złożone ze skończenie wielu sympleksów (punktów, odcinków, trójkątów, czworokątów, . . .) posklejanych porządnie, czyli wspólnymi ścianami. Kilka przykładów można obejrzeć na rysunkach na marginesie. Taki kompleks jest obiektem topologicznym, ale równocześnie można go zdefiniować kombinatorycznie – za pomocą skończonego zbioru informacji. Wystarczy umieścić w odpowiednio dużej przestrzeni \mathbb{R}^n wszystkie wierzchołki wszystkich sympleksów, które mają się znaleźć w kompleksie, i powiedzieć, które pary łączymy odcinkiem, potem które z powstałych w ten sposób trójkątów mają być wypełnione, itd. Właśnie jako takie zestawy danych możemy traktować zbiory \mathcal{I} i \mathcal{O} . Każdemu z nich przypisujemy kompleks, którego wierzchołkami są wektory zawierające tylko jedną wartość różną od \perp (na rysunkach będą one oznaczane przez nazwę procesu, któremu odpowiada wartość określona, i tę wartość). Odcinek łączący dwa wierzchołki należy do kompleksu, jeśli w zbiorze znajduje się wektor, który ma dokładnie dwie wartości różne od \perp – takie, jak w wektorach będących wierzchołkami, i na tych samych miejscach. Analogicznie dodajemy do kompleksu sympleksy wyższych wymiarów – odpowiadają one wektorom z mniejszą liczbą \perp . Rysunki 3 i 4 przedstawiają kompleksy wejściowe i wyjściowe dla zadań *Binary Consensus* i *Quasi-Consensus*. Przekształcenie Δ przypisuje sympleksom z kompleksu \mathcal{I} zbiory sympleksów z \mathcal{O} , zachowując wymiar. Wobec tego, niestety, zazwyczaj nie może być ono przekształceniem symplecjajnym, czyli takim, że obrazem każdego sympleksu jest sympleks, ale okaże się, że czasem jest w pewnym sensie bliskie takiemu.

Przyda się nam jeszcze jedna operacja na kompleksie symplecjajnym – podział kompleksu. Każdy sympleks wyjściowego kompleksu możemy rozbić na kilka mniejszych (dodając nowe wierzchołki), nawet na wiele różnych sposobów, ale żeby wynikiem był kompleks symplecjajny, rozbiecie musi spełniać warunek dobrego sklepania sympleksów. Łatwo narysować, na przykład, podział barycentryczny, który powstaje przez dodanie i odpowiednie połączenie środków ciężkości wszystkich sympleksów. Najniższy kompleks na rysunku 2 to właśnie podział barycentryczny

trójkąta. Nośnikiem sympleksu w podziale nazywamy najmniejszy sympleks wyjściowego kompleksu, który go zawiera.

Czasami wierzchołki w kompleksie są pokolorowane; jak zwykle wymagamy, żeby wierzchołki połączone odcinkiem miały różne kolory. Wtedy, konstruując podział, chcemy, żeby kolorowanie można było rozszerzyć na dodane wierzchołki, i wybieramy pewne rozszerzenie. Na kompleksach \mathcal{I} i \mathcal{O} uwzględniamy kolorowanie pochodzące od oznaczeń wierzchołków – kolory to pary złożone z nazwy procesu i wartości wejściowej lub wyjściowej.

Te pojęcia pozwolą nam przyjrzeć się twierdzeniu, które opisuje zależność między istnieniem dobrego rozwiązania dla zadania a strukturą odpowiadających mu kompleksów symplecjajalnych. Ale jakie rozwiązanie uznamy za dobre? Wymagamy oczywiście, żeby wyniki były zgodne z funkcją Δ . Dodatkowo protokół, czyli algorytm wykonywany przez procesy, nie może pozwalać na nieskończone oczekiwanie – każdy proces po skończonej liczbie kroków musi zakończyć działanie, poprawnie lub z błędem (ang. *wait-free protocol*). Żeby można było tę definicję wykorzystać w ścisłym dowodzie, trzeba ją zapisać w języku automatów, ale my poprzestaniemy na intuicji.

Poniższe twierdzenie pochodzi z pracy Maurice'a Herlihy'ego i Nira Shavita, *The Topological Structure of Asynchronous Computability (Struktura topologiczna obliczeń rozproszonych)*, z 1999 roku. Jak widać, pomysł nie jest stary, i można się spodziewać, że nie został jeszcze całkiem wyeksploatowany. . .

Twierdzenie 1. *Dla zadania $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ istnieje protokół bez oczekiwania wtedy i tylko wtedy, gdy istnieje podział $\sigma(\mathcal{I})$ kompleksu \mathcal{I} (z opisanym wyżej kolorowaniem) oraz przekształcenie symplecjajalne $\mu : \sigma(\mathcal{I}) \rightarrow \mathcal{O}$ zachowujące kolorowanie i takie, że dla każdego wierzchołka $s \in \sigma(\mathcal{I})$ punkt $\mu(s)$ leży w obrazie nośnika (s, \mathcal{I}) przy funkcji Δ .*

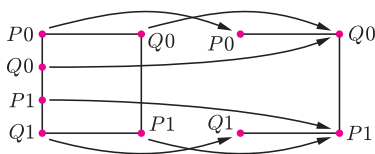
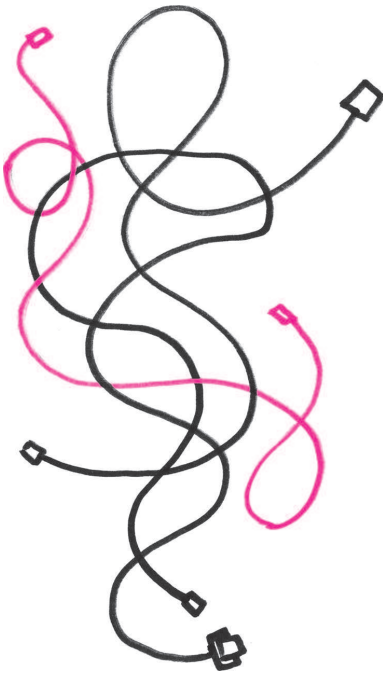
To oznacza, że jeśli chcemy udowodnić istnienie dobrego protokołu dla pewnego zadania, musimy znaleźć podział kompleksu wejściowego uwzględniający naturalne kolorowanie oraz odwzorowanie μ z tego podziału w \mathcal{O} , zgodne z warunkami zadania, czyli z funkcją Δ . Przyjrzyjmy się, jak to wygląda dla zadania *Quasi-Consensus* (rys. 5).

Kompleksy \mathcal{I} i \mathcal{O} już znamy, a $\sigma(\mathcal{I})$ tworzymy, dodając na krawędzi $(P0, Q1)$ dwa punkty, o kolorach $P1$ i $Q0$. Przekształcenie μ przeprowadza krawędzie $(P0, Q0)$, $(P1, Q0)$, $(P1, Q1)$ na ich odpowiedniki w kompleksie wyjściowym. Trzy odcinki składające się na krawędź $(P0, Q1)$ również przechodzą przy μ na te krawędzie, zgodnie z kolorami wierzchołków. Łatwo sprawdzić, że warunek zgodności z Δ jest spełniony. Stąd dla zadania *Quasi-Consensus* istnieje protokół bez oczekiwania.

A co z zadaniem *Binary Consensus*? Spójrzmy jeszcze raz na rysunek 3 – kompleks wyjściowy jest niespójny. Warunek zgodności z Δ wymaga, żeby obie składowe \mathcal{O} były w obrazie μ . Ale μ musi być ciągle, a \mathcal{I} jest spójny, co prowadzi do sprzeczności. Dla tego zadania dobry protokół nie istnieje. A żeby to wykazać, wystarczyło przyjrzeć się bardzo pobieżnie strukturze topologicznej kompleksów!

Można stawiać zarzuty, że twierdzenie daje wnioski czysto egzystencjalne – a co nam przyjdzie z wiedzy, że pewien program można napisać, jeśli nie wiemy, jak to zrobić? W rzeczywistości jest lepiej, niż mogłoby się na pierwszy rzut oka wydawać. Co prawda z samego twierdzenia nie możemy wiele wnioskować o postaci protokołu, ale dowód jest konstruktywny. Jego analiza pozwala na zrozumienie i zapisanie odpowiedniego kodu, chociaż zwykle nie w takiej postaci, jaka wydawałaby się najbardziej naturalna.

Użyliśmy twierdzenia 1 tylko do rozstrzygnięcia dwóch bardzo prostych problemów. Ale można je stosować do istotnie bardziej złożonych sytuacji i otrzymywać ważne rezultaty. To twierdzenie pozwoliło, między innymi, na zbadanie zadania *Renaming*. W pewnych przypadkach, dla zbioru wyników niewiele większego od zbioru procesów, istnienie protokołu bez oczekiwania było wcześniej nierozstrzygnięte. Do badania zadań trudniejszych niż opisane przykłady często potrzebne są bardziej subtelne własności topologiczne kompleksów \mathcal{I} i \mathcal{O} niż tylko spójność. Przydaje się, na przykład, lemat Spernera oraz różne metody wykrywania dziur w przestrzeniach: grupy homotopii, w szczególności grupa podstawowa i grupy homologii, które dla kompleksów symplecjajalnych nietrudno obliczyć. Możliwe, że topologia ma jeszcze dużo do powiedzenia w informatyce. . .



Rys. 5. Działanie przekształcenia μ dla zadania *Quasi-Consensus*.

W tym miejscu warto sobie przypomnieć o problemie sekcji krytycznej i sygnalizowanych trudnościach w jego ominięciu. . .

Kosmiczna linijka

10. Wielki Atraktor: dokąd płyną galaktyki?

Odległość 147 mln lat świetlnych (40 Mpc na linijce)

Galaktyki we Wszechświecie nie są rozmieszczone równomiernie, lecz tworzą ogromne struktury, nieco na kształt nieporządanej pajęczej sieci. W jej „oczkach” rezydują bogate gromady galaktyk, wzdłuż „nitki” układają się większe struktury zwane supergromadami, a pomiędzy nimi znajdują się gigantyczne pustki. Gromady galaktyk to skupiska bardzo zwarte i stosunkowo łatwe do zaobserwowania. Znane są od czasów katalogu Palomarskiego Przeglądu Nieba, na podstawie którego w 1958 roku George O. Abell opublikował dane dla ponad 4 tys. gromad. Odkrycie większych struktur nie było łatwe, ponieważ nie wyróżniają się one aż tak wyraźnie na niebie. Charakterystyczną wielkością, definiującą istnienie struktury wielkoskalowej, jest tak zwany kontrast gęstości. Wynosi on jedynie około 0,5–2, co oznacza, że w obszarach o podwyższonej gęstości jest dwukrotnie więcej galaktyk, w obszarach zaś o obniżonej gęstości dwukrotnie mniej niż średnio we Wszechświecie.

Aczkolwiek pierwsze sugestie o istnieniu supergromad wysunięto już w latach 1930. (Clyde Tombaugh, odkrywca Plutona, twierdził, że w Perseuszu znajduje się supergromada 2 tysięcy galaktyk), to jednak początkowo nie dawano im wiary. Dopiero lata 70., kiedy zaczęto wykonywać systematycznie wielkie przeglądy nieba, potwierdziły istnienie supergromad. Supergromadę Lokalną, w której znajduje się Grupa Lokalna Galaktyk i należąca do niej Droga Mleczna, jako pierwszy zbadał i opisał francuski astronom Gerard de Vaucouleurs. Trójwymiarowa mapa tej supergromady, sporządzona przez Tully’ego i Fishera, pokazała, że aż 98% widocznych galaktyk zajmuje jedynie 5% objętości tej supergromady – a więc pozostały obszar to wielka pustka.

Wielki Atraktor jest najśłynniejszą z wielkich struktur. Jest to skupisko gromad galaktyk, znajdujące się w kierunku gwiazdozbiorów Centaura i Hydry, na południowej półkuli nieba. Rozmiar tego obszaru to około 400 milionów lat świetlnych. Zawiera on około 100 000 galaktyk o łącznej masie około $5 \cdot 10^{16}$ mas Słońca (masy podawane przez różnych autorów różnią się znacząco, ponieważ oceny oparte na przyciąganiu grawitacyjnym komplikuje fakt obecności innej gęstej struktury – większej, położonej dalej, ale w podobnym kierunku, o nazwie Supergromada Shapleya).

Centralną część Wielkiego Atraktora stanowi widziana z boku silnie spłaszczona supergromada (znana pod angielską nazwą *Centaurus Wall*), a jądro tej struktury to niezwykle bogata gromada galaktyk o nazwie Abell 3627, znana także jako gromada Norma.

Po raz pierwszy obecność tajemniczego atraktora zauważono w latach 70. ubiegłego wieku podczas badania rozkładu prędkości swoistych dla pobliskich galaktyk. Okazało się, że lokalnie występują dosyć duże odstępstwa od prawa Hubble’a, mówiącego o ogólnej ucieczce galaktyk. Zgodnie z tym prawem każda galaktyka oddala się od dostatecznie odległych, i to tym szybciej, im dalej się od nich znajduje. O oddalaniu się galaktyki świadczy fakt przesunięcia ku czerwieni jej linii widmowych.

Wielki Atraktor przyciąga gromady galaktyk z naszego sąsiedztwa, sprawia też, że Droga Mleczna zamiast przybliżać się do centrum gromady galaktyk w Pannie (do której należy), oddala się od niego. Nie oznacza to jednak, że zbliżamy się do Wielkiego Atraktora. Tak naprawdę, w sensie bezwzględnym oddalamy się od niego z prędkością około 4000 km/s, ale o 627 km/s wolniej, niż gdyby nie było jego oddziaływania grawitacyjnego.

Istnienie Wielkiego Atraktora potwierdziła w 1988 roku kierowana przez Donalda Lyndena-Bella grupa astronomów, znana jako „Siedmiu Samurajów”. Nie było to łatwe, ponieważ Wielki Atraktor leży bardzo blisko płaszczyzny naszej Galaktyki (szerokość galaktyczna 10°). Znajdujące się w Drodze Mlecznej pył i gaz międzygwiazdowy oraz gwiazdy przesłaniają nam, niestety, widoczność, przez co powstaje wrażenie, jakby odległe galaktyki „unikaly” tej strefy. W dodatku kontrast gęstości w całym obszarze Atraktora to zaledwie 1,1, przez co jego lokalizacja oparta była przede wszystkim na wyznaczonych kierunkach prędkości swoistych galaktyk, a potwierdzające ją mapy rozkładu gęstości zostały wykonane dopiero w połowie lat 90. O ile bowiem trudno jest zliczać galaktyki i ocenić, czy na pewno powodem ruchu Grupy Lokalnej jest właśnie Wielki Atraktor, o tyle pomiar prędkości jest prosty i dokładny, ponieważ tę absolutną prędkość kosmiczną mierzymy w odniesieniu do mikrofalowego promieniowania tła, które wypełnia cały Wszechświat.

Bożena CZERNY, Agnieszka JANUOK

Konkurs zadań astronomicznych

Na rozwiązania zadań A 19 i A 20 czekamy do 1 listopada 2009 r. (decyduje data stempla pocztowego) pod adresem:

Centrum Astronomiczne
im. Mikołaja Kopernika
ul. Bartycka 18
00-716 Warszawa

z dopiskiem na kopercie „Konkurs Deltę”.

A 19. Jak długo trwałby spadek swobodny na Ziemię z wysokości (orbity) Księżyca? Rozmiar Ziemi i eliptyczność orbity Księżyca zaniedbać. [2 pkt]

A 20. Piloci samolotu lecącego dokładnie na zachód (po równoleżniku) z prędkością 1000 km/h widzą Słońce nieruchome względem samolotu. Jaka jest szerokość geograficzna φ trasy? [1 pkt]

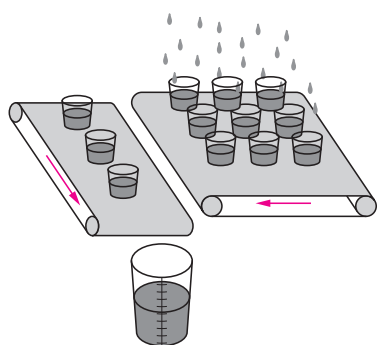
Rozwiązania zadań z numeru 8/2009

A 15. Masa Ziemi wynosi $M_Z = 5,97 \cdot 10^{24}$ kg, $G = 6,67 \cdot 10^{-11}$ m³kg⁻¹s⁻², a $c = 3 \cdot 10^8$ m/s. Zatem $R_{BH} = \frac{2GM_Z}{c^2}$. Podstawiając wartości liczbowe otrzymujemy $R_{BH} = 8,86 \cdot 10^{-3}$ m.

A 16. Robocik może jechać z taką prędkością, by w ciągu przemieszczenia sygnału na Ziemię i z powrotem nie przejechać więcej niż zasięg ostrego widzenia kamery. Tak więc $v_r = \frac{d}{2t_p}$, stąd $v_r = 0,015$ km/h = 0,41 cm/s!

Kamery CCD w astronomii

Wojciech PYCH*



Schemat urządzenia mechanicznego działającego analogicznie do CCD.

Samo określenie CCD (po angielsku *charge-coupled device*) oznacza jedynie zestaw miniaturowych kondensatorów połączonych w taki sposób, że możliwe jest przesłanie ładunku zgromadzonego w pojedynczym elemencie do jednego z jego sąsiadów. Na końcu takiego łańcucha ładunek z każdego elementu można zamienić na napięcie elektryczne, a następnie na sygnał cyfrowy. Urządzenia CCD mogą więc być wykorzystywane jako rodzaj pamięci elektronicznej lub jako elementy opóźniające sygnał w układach elektronicznych. Początkowo układy te konstruowane były w nadziei na wykorzystanie ich jako pamięci do komputerów. Okazało się jednak, że efekt fotoelektryczny może być źródłem elektronów gromadzonych w kondensatorach matrycy CCD, dzięki czemu ich ilość jest proporcjonalna do natężenia padającego światła. Odpowiednie rozmieszczenie elementów światłoczułych pozwala więc na cyfrową rejestrację padającego na nie światła, a zatem obrazu.

Na rysunku przedstawiamy schemat urządzenia mechanicznego działającego analogicznie do kamer CCD. W tym przypadku na przesuującym się taśmociągu umieszczone zostały kubeczki zbierające krople padającego deszczu. Na jego końcu woda z kubeczków przelewana jest do jednego rzędu kubeczków umieszczonego na taśmociągu przesuującym się w kierunku prostopadłym. Na końcu drugiego taśmociągu znajduje się naczynie ze skalą pozwalającą zmierzyć ilość wody w kolejnych kubeczkach.

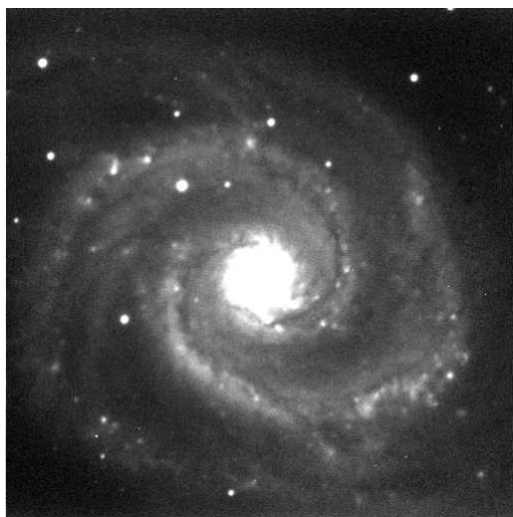
Technologia CCD ma już 40 lat. Pierwsza działająca matryca CCD powstała w 1969 roku w Bell Telephone Laboratories. Początki nie były spektakularne. Pierwsze urządzenie miało zaledwie 9 elementów ustawionych w rzędzie. Nie rejestrowało ono światła, a jedynie umożliwiała przesłanie ładunku z jednego końca na drugi. Główni konstruktorzy – Willard Boyle i George Smith – ogłosili to wydarzenie w artykule opublikowanym w kwietniu 1970 roku. Już wtedy zwrócili uwagę na potencjalną możliwość wykorzystania matrycy CCD do rejestracji obrazów. Ostatecznie CCD nigdy nie były wykorzystywane jako pamięci do komputerów, za to skutecznie zastąpiły klisze fotograficzne.

Pierwsza matryca CCD, która w praktyce służyła do obrazowania, została wyprodukowana przez firmę Fairchild Electronics w 1974 roku. Była to mozaika jedynie 100×100 punktów obrazu (pikseli). Już wtedy podjęto próby wykorzystania jej do pierwszych cyfrowych zdjęć astronomicznych. Jednym z podstawowych parametrów urządzeń światłoczułych jest wydajność kwantowa, czyli stosunek liczby fotonów zarejestrowanych do liczby fotonów padających. Dla nowego instrumentu wydajność kwantowa wynosiła około 0,5% i była nieco niższa od tego, co oferowały najlepsze płyty fotograficzne. Innym ważnym parametrem jest szum odczytu sygnału. W matrycy z 1974 roku wynosił on 30 elektronów. Oznacza to, że na obraz ciemnego tła nieba nałożona była „kaszka” o takim rozrzucie zliczeń w sąsiednich punktach.

Pierwsza kamera CCD, na stałe zamontowana przy profesjonalnym jednometrowym teleskopie w obserwatorium Kitt Peak w Arizonie, uruchomiona została w 1979 roku. Zawierała ona matrycę firmy RCA o rozdzielczości 512×320 pikseli. Wtedy też ujawniła się przewaga nowych detektorów nad używanymi w tym czasie płytami fotograficznymi. Wydajność kwantowa okazała się około 50-krotnie wyższa (w barwie czerwonej), a sygnał liniowo zależał od oświetlenia. Początkowo największą słabością była bardzo mała rozdzielczość w porównaniu do dużych płyt fotograficznych.

W Polsce po raz pierwszy kamera CCD została uruchomiona w Obserwatorium Astronomicznym Uniwersytetu Warszawskiego. Współpracuje ona z 60-centymetrowym teleskopem w stacji obserwacyjnej w Ostrowiku. Rozdzielczość obrazów produkowanych przez tę kamerę to 512×512 pikseli.

* Centrum Astronomiczne
im. Mikołaja Kopernika, Warszawa



Zdjęcie galaktyki spiralnej M51 – jedno z pierwszych zdjęć wykonanych za pomocą pierwszej w Polsce astronomicznej kamery CCD (styczeń 1992).

Na zdjęciu pokazana jest galaktyka spiralna M51. Jest to jedno z pierwszych zdjęć nieba wykonanych tą kamerą.

Od momentu powstania matryc CCD do dzisiaj są one systematycznie ulepszane. Obecnie można już kupić matryce o rozdzielczościach rzędu $10\,000 \times 10\,000$ pikseli. Dostępny dla różnych rodzajów kamer zakres długości fal elektromagnetycznych zaczyna się od 0,1 nm (promienie X) i ciągnie się aż do 1100 nm (podczerwień). Mamy też specjalne detektory cząstek oparte na tej technologii. Wydajność transferu ładunku pomiędzy sąsiednimi elementami matrycy wzrosła do 99,999%, a szum odczytu spadł do 1 elektronu. Zakres dynamiki, czyli stosunek zliczeń w najjaśniejszym, nieprześwietlonym elemencie obrazu do zliczeń na poziomie sygnału własnego kamery, osiąga wartość 10^6 .

Współcześnie detektory cyfrowe praktycznie całkowicie wyparły tradycyjne metody fotograficzne z działalności naukowej. Zastosowanie kamer CCD w astronomii stało się źródłem ogromnego postępu. Dzięki nim nasze teleskopy uzyskały większy zasięg zarówno pod względem jasności obserwowanych obiektów, rozdzielczości kątowej uzyskiwanych obrazów, jak też dokładności pomiarów jasności i pozycji kosmicznych źródeł promieniowania. Gdyby

astronomia dalej opierała się na fotografii analogowej, nie mielibyśmy szans na wiele dokonanych już odkryć. Bez kamer CCD nie byłoby chociażby ogromnego przecież wkładu Teleskopu Hubble'a do naszej wiedzy o Wszechświecie.

Matryce CCD są pierwszymi, chociaż obecnie już nie jedynymi urządzeniami umożliwiającymi cyfrową rejestrację obrazu. W życiu codziennym jesteśmy świadkami gwałtownego wypierania filmów fotograficznych przez technologię cyfrową.



Zadania

Redaguje Ewa CZUCHRY

F 749. Niebezpiecznie jest fotografować tygrysa z odległości mniejszej niż 20 m. Jakich rozmiarów powinna być *camera obscura* z otworkiem o średnicy $d = 1$ mm, aby widoczne były pręgi tygrysa? Odległość między pręgami na sierści tygrysa wynosi $l = 20$ cm.

Rozwiązanie na str. 13

F 750. Na środku płaskiego ekranu znajduje się punktowe źródło światła. Równoległe do ekranu umieszczono zwierciadło płaskie w kształcie trójkąta równobocznego o długości boku $a = 10$ cm. Oszacować rozmiar S „zajęczka” na ekranie.

Rozwiązanie na str. 24

Redaguje Waldemar POMPE

M 1255. Niech a, b, c, d oraz n będą liczbami całkowitymi dodatnimi spełniającymi zależność

$$a^2 + b^2 + c^2 + d^2 = 7 \cdot 4^n.$$

Wykazać, że każda z liczb a, b, c, d jest większa lub równa 2^{n-1} .

Rozwiązanie na str. 16

M 1256. Liczby rzeczywiste a_1, a_2, \dots, a_n , gdzie $n \geq 4$, spełniają nierówności

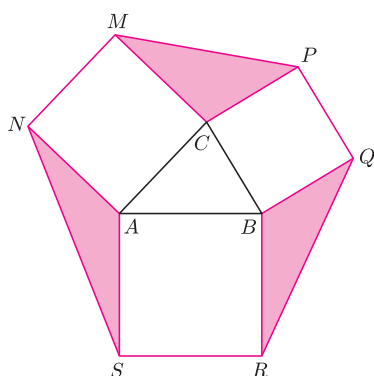
$$a_1 + a_2 + \dots + a_n \geq n \quad \text{oraz} \quad a_1^2 + a_2^2 + \dots + a_n^2 \geq n^2.$$

Udowodnić, że co najmniej jedna z liczb a_1, a_2, \dots, a_n jest większa lub równa 2.

Rozwiązanie na str. 16

M 1257. Na bokach AB, BC, CA trójkąta ABC zbudowano – na zewnątrz trójkąta ABC – kwadraty $ABRS, BCPQ$ oraz $ACMN$. Dowieść, że pola trójkątów NAS, BRQ oraz MPC są równe.

Rozwiązanie na str. 24



*student, Wydział Matematyki
i Informatyki, Uniwersytet Wrocławski

Abstrakcja (czyli uproszczenie) jest w informatyce wszechobecna. Mając jakiś obiekt, wiemy zazwyczaj, co powinniśmy mu wprowadzić na wejście i czego spodziewać się na wyjściu. Takie informacje nas już w zupełności satysfakcjonują. Możemy używać tego obiektu, stroniąc od szczegółów jego budowy. I tak, tworząc procesor, możemy pominąć to, z czego są zbudowane bramki logiczne. Projektując system operacyjny, nie musimy przejmować się szczegółami budowy procesora, a pisząc zwykły program, nie wnikamy w szczegóły działania systemów operacyjnych. Oczywiście, takie podejście ma mnóstwo zalet. Ma też jednak pewną wadę.

Rozważmy następujący program będący tabliczką mnożenia. To, w jakiej kolejności będziemy obliczać poszczególne wartości, nie powinno mieć w sumie żadnego znaczenia. W szczególności, żadnej różnicy nie powinniśmy odczuć, zamieniwszy miejscami pętle w programie. Program nadal wykona takie same obliczenia, taką samą liczbę razy, tylko w innej kolejności.

```
#define N (1 << 13)

int tab[N][N];

int main()
{
    for (int j = 0; j < N; j++)
        for (int i = 0; i < N; i++)
            tab[i][j] = i*j;
    return 0;
}
```

Program 1: Tabliczka mnożenia

Jednak okazuje się, że kolejność pętli w programie ma znaczenie. Na komputerze średniej klasy program 1 wykonywał się przez 7,93 s. Z kolei po zamianie pętli miejscami przyspieszył do 1,32 s. Zysk ponad sześciokrotny!

Drugi przykład. Mamy program 2, który w pętli oblicza sumy dwudziestu elementów tablicy. Zastanówmy się, co się stanie, gdy linijkę, w której deklarujemy tablicę `tab`, zamienimy na `int tab[N][M+512];`? Teoretycznie nie powinno to spowodować żadnych zmian – program wciąż wykona taką samą liczbę instrukcji. Wprawdzie zadeklarowaliśmy trochę więcej pamięci, jednak procesor na tę dodatkową pamięć nawet nie spojrzy. Czy aby na pewno?

```
#define N 100
#define M (1 << 19)

int tab[N][M];

int main()
{
    for (int i = 0; i < N-20; i++)
        for (int j = 0; j < M; j++)
            tab[i][j] = tab[i+1][j] + tab[i+2][j] + ... + tab[i+20][j];
    return 0;
}
```

Program 2: Sumowanie tablicy

Ponownie przyjrzyjmy się czasom działania obu wersji programu. W pierwszej wersji program działa 13,16 s, a po powiększeniu tablicy – o blisko 10 sekund szybciej, tj. 3,18 s.

Skąd wynikają te anomalie czasowe? Z powodu abstrakcji! Podczas omawiania powyższych programów nie zastanawialiśmy się w ogóle, jak procesor interpretuje poszczególne instrukcje i co jest dla niego „wygodniejsze”. Skupiliśmy się jedynie na tym, że w obu programach wykonywane są takie same instrukcje, i to tyle samo razy. Przyjrzymy się teraz budowie i działaniu pamięci cache, aby lepiej zrozumieć, czym różnią się omawiane przez nas przykłady.

W przeciętnym komputerze możemy wyróżnić kilka rodzajów pamięci. Mamy dyski twarde, pamięć RAM oraz pamięć podręczną procesora (tzw. pamięć cache). Te typy pamięci różnią się rozmiarem, ceną oraz czasem dostępu. Dyski twarde są ogromne; za jeden gigabajt płacimy parę złotych, a czas dostępu jest rzędu 10 ms = 10 000 000 ns (nanosekund). Pamięć RAM jest już dużo mniejsza niż dyski twarde; za jeden GB zapłacimy kilkadziesiąt złotych, a czas dostępu do pamięci jest rzędu 60 ns. Pamięć cache procesora jest mała; za jeden gigabajt tego rodzaju pamięci zapłacimy kilkanaście tysięcy złotych. Za to czas dostępu do tego rodzaju pamięci jest rzędu kilku nanosekund.

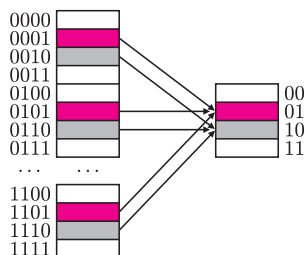
Dążeniem twórców sprzętu jest taki model komputera, w którym średnia cena za gigabajt pamięci w całym komputerze byłaby zbliżona do ceny gigabajtu dysku twardego, natomiast średni czas dostępu do pamięci byłby rzędu czasu dostępu do pamięci cache. Cel ten, jak łatwo się domyślić, jest – póki co – niemożliwy do uzyskania, jednak nasze komputery są bliżej tego modelu, niż mogłoby nam się wydawać.

W cache'u nie możemy zmieścić pamięci całego programu. Pojawia się pewien problem – gdy procesor potrzebuje użyć jakiejś zmiennej, to powinna się ona już znajdować w pamięci cache. Zatem maszyna musi zabawić się w jasnovidza i spróbować przewidzieć, jakie zmienne będą przez program użyte w najbliższej przyszłości. Na szczęście komputerowi z pomocą przychodzi zasada lokalności. Mówi ona, że jeśli jakiś adres pamięci jest właśnie używany, to prawdopodobnie za chwilę będzie znowu potrzebny – albo on, albo adresy znajdujące się blisko niego. Przykładami mogą być listy rozkazów (po wykonaniu rozkazu prawdopodobnie będzie wykonywany rozkaz znajdujący się bezpośrednio po nim) albo tablice (jeśli jakiś element został wykorzystany, to za chwilę prawdopodobnie będziemy potrzebować sąsiedniego elementu).

Ogólna zasada działania pamięci cache jest następująca. Procesor prosi o jakiś adres pamięci. Pamięć cache sprawdza, czy taki adres zawiera. Jeśli tak, to przesyła go do procesora. Jeśli nie, to ściąga go z pamięci RAM wraz z całym blokiem (pamięć RAM jest podzielona na mniejsze porcje informacji, zwane blokami), a następnie przesyła procesorowi potrzebne dane.

Opiszemy teraz, jak bloki umieszczane są w pamięci cache. Zazwyczaj używane jest jedno z tzw. mapowań: skojarzeniowe, sekcyjne i sekcyjno-skojarzeniowe, przy czym w praktyce najczęściej ostatnie. My jednak skupimy się na mapowaniu sekcyjnym, gdyż jest do niego bardzo podobne, a zarazem nieco łatwiejsze do opisanego.

Pamięć cache jest podzielona na sekcje – są to odpowiedniki bloków w pamięci RAM. Każdemu blokowi RAM-u przyporządkowana jest jedna sekcja w pamięci cache, wyznaczona przez jego adres modulo liczba sekcji (rys. 1), która to liczba jest zawsze potęgą dwójki. Dzięki temu, aby „obliczyć”, jaki adres sekcji przyporządkowaliśmy danemu blokowi, nie potrzebujemy wykonywać żadnych pracochłonnych operacji – wystarczy spojrzeć na ostatnie bity adresu bloku. Ponadto, operacja modulo gwarantuje nam, że kilka sąsiednich bloków pamięci RAM może się w tym samym czasie znajdować w pamięci cache (co jest korzystne ze względu na zasadę lokalności). Z drugiej strony może się, oczywiście, zdarzyć, że kilku blokom jest przyporządkowana ta sama sekcja – wówczas w momencie sprowadzania kolejnego z tych bloków do cache'u, poprzedni zostanie nadpisany.

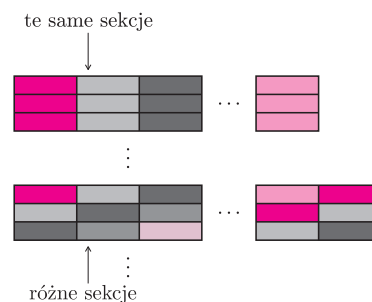


Rys. 1

Po tej nieco dokładniejszej analizie mechanizmu działania pamięci podręcznej procesora przyjrzyjmy się

ponownie naszym programom. W pierwszym z nich istotną rolę niespodziewanie odgrywała kolejność pętli. Aby zrozumieć, skąd wzięła się tak duża różnica, musimy sprawdzić, w jakiej kolejności w pamięci ułożone są kolejne elementy tablicy dwuwymiarowej. Nietrudno sprawdzić (na przykład pisząc prosty program w języku C/C++), że tablica taka zajmuje spójny obszar pamięci. Ułożona jest w następujący sposób: `tab[0][0], tab[0][1], tab[0][2], ..., tab[0][N-1], tab[1][0], ...` itd. W takiej kolejności przeglądamy tablicę, jeżeli odwrócimy kolejność pętli w programie 1. Dzięki temu działa zasada lokalności i procesor łatwo przewiduje, które adresy pamięci będą za chwilę potrzebne. W oryginalnym programie 1 przeglądamy tablicę w sposób dla procesora bardzo chaotyczny. Komórka `tab[0][0]` w ogóle nie leży w pobliżu `tab[1][0]`, przez co procesor, próbując przewidzieć, jakie kolejne adresy będą wykorzystywane, nieustannie się myli, więc musi przysyłać z pamięci RAM do pamięci cache ciągle nowe adresy – a to ma negatywny wpływ na czas działania programu.

Drugi program wydaje się jeszcze ciekawszy. Faktycznie, w pierwszym programie w celu usprawnienia zmieniliśmy kolejność wykonywania instrukcji, natomiast w tym przypadku nawet tego nie zrobiliśmy! Sekret tkwi w tym, że w pierwszej wersji programu 2 drugi wymiar tablicy był potęgą dwójki (choć tak naprawdę ważne jest tylko to, że był on wielokrotnością rozmiaru pamięci cache mojego procesora) – patrz rysunek 2. Dlatego komórki `tab[i+1][j], tab[i+2][j], tab[i+3][j], ..., tab[i+20][j]` trafiały zawsze do tej samej sekcji w pamięci podręcznej. Przebieg tego programu wyglądał więc następująco. Aby obliczyć `tab[i][j]`, procesor musi dodać komórki `tab[i+1][j], ..., tab[i+20][j]` – prosi więc pamięć cache najpierw o `tab[i+1][j]`. Pamięć cache jej nie ma, więc ściąga z pamięci RAM. Następnie procesor prosi o `tab[i+2][j]`. Pamięć cache jej nie ma (w sekcji, w której powinna być ta komórka, jest `tab[i+1][j]`), więc pobiera ją z pamięci głównej, i tak dalej aż do `tab[i+20][j]`. Następnie procesor musi obliczyć `tab[i][j+1]`, co znowu wymagać będzie dwudziestu ściągnięć bloku do pamięci cache itd.



Rys. 2

Gdy zwiększymy liczbę kolumn w tablicy o długość bloku, zagwarantujemy, że komórki `tab[i+1][j], tab[i+2][j], ..., tab[i+20][j]` będą trafiać do sąsiednich sekcji w pamięci podręcznej.

I choć przy obliczaniu pierwszej sumy będzie trzeba wszystkie te bloki ściągnąć do pamięci cache, to przy obliczaniu sumy $\text{tab}[i+1][j+1], \text{tab}[i+2][j+1], \dots, \text{tab}[i+20][j+1]$ odpowiednie bloki będą się już znajdować w pamięci podręcznej, dzięki czemu program nie straci cennych nanosekund.

Być może ktoś powie, że nie warto się w to wszystko w ogóle zagłębiać, gdyż przyspieszamy program jedynie o stały czynnik. Jednak nakład pracy, jaki włożyliśmy w zmianę programów, był minimalny. Jeśli mamy program działający 10 godzin, a potrzebujemy otrzymać wynik w ciągu godziny, to owszem – możemy kupić

dziesięciokrotnie szybszy komputer. Ale czasami możemy zaoszczędzić dużo pieniędzy, po prostu zamieniając kilka linijek miejscami i dodając kilka pozornie nieistotnych znaków w odpowiednich miejscach programu.

Na koniec małe ćwiczenie dla Ciebie, Drogi Czytelniku. Otwórz swoją ulubioną wyszukiwarkę internetową i wyszukaj „mnożenie macierzy C++”. Wejź na pierwszą lepszą stronę. Czy implementacja, którą znalazłeś, jest napisana optymalnie z punktu widzenia działania mechanizmu cache? Czy można ją przyspieszyć, używając metod opisanych w tym artykule? Jak to zrobić?

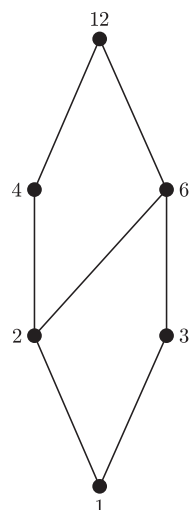
Co to znaczy zrekonstruować porządek?

Marta PRZYBOROWSKA*

W tym artykule przedstawimy problem rekonstrukcji zbioru częściowo uporządkowanego. Będziemy rozważać jedynie zbiory skończone. Do sformułowania problemu będziemy potrzebować kilku podstawowych pojęć.

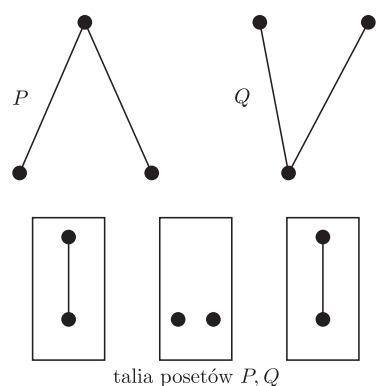
Porządkiem częściowym na zbiorze X nazywamy relację $<$ spełniającą własność przechodniości i przeciwzwrotności. Mówiąc, że relacja $<$ jest *przechodnia*, mamy na myśli, że jeżeli a, b, c są takimi elementami zbioru X , że $a < b$ i $b < c$, to również $a < c$. *Przeciwzwrotność* relacji $<$ oznacza, że nigdy nie zachodzi $a < a$. Z tych aksjomatów wynika, że również nigdy nie zachodzi jednocześnie $a < b$ i $b < a$. Jeżeli relacja $<$ jest porządkiem częściowym na zbiorze X , to parę $(X, <)$ nazywamy *zbiorem częściowo uporządkowanym* lub krótko *posetem* (ang. *partially ordered set*). Zbiór liczb całkowitych $X = \{1, 2, 3, 4, 6, 12\}$, w którym $a < b$ oznacza, że a dzieli b i $a \neq b$, jest bez wątpienia przykładem posetu.

Mówimy, że element b *pokrywa* element a w posecie $(X, <)$, jeżeli $a < b$ i dla żadnego elementu $c \in X$ nie zachodzi $a < c < b$. *Diagramem* posetu $P = (X, <)$ nazywamy graf skierowany $D(P)$ na zbiorze wierzchołków X , w którym rysujemy łuk od a do b wtedy i tylko wtedy, gdy b pokrywa a . Rysując diagram posetu, kierujemy krawędzie w górę, tak aby otrzymać charakterystyczny obraz porządku wyróżniający odpowiednie poziomy elementy zbioru X . Wówczas $a < b$ wtedy i tylko wtedy, gdy istnieje w diagramie $D(P)$ ścieżka wznosząca się w górę od a do b . Na rysunku 1 widzimy diagram posetu dzielników liczby 12.



Rys. 1

Zauważmy, że diagram posetu można narysować na wiele sposobów, otrzymując różne rysunki grafu $D(P)$. Posety P i Q nazywamy *izomorficznymi*, jeżeli ich diagramy można tak narysować, że otrzymane rysunki będą identyczne. Izomorfizm posetów P i Q oznaczamy przez $P \simeq Q$.



Rys. 2

Sformułowanie problemu

Niech $P = (X, <)$ będzie posetem i niech x będzie dowolnym elementem zbioru X . Niech P_x oznacza poset otrzymany przez usunięcie elementu x ze zbioru X . Przypuśćmy, że każdy z diagramów $D(P_x)$, $x \in X$, został narysowany na osobnej *karcie*, ale bez oznakowania wierzchołków. Kolekcję wszystkich takich kart posetu P nazywamy jego *talą*. Zwróćmy uwagę, że talia może zawierać wiele identycznych kart, a jej liczność jest równa liczbie elementów zbioru X .

Czy mając daną talię kart pewnego posetu, możemy *zrekonstruować* go z dokładnością do izomorfizmu?

Innymi słowy, pytamy, czy jest możliwe, aby dwa różne (czyli nieizomorficzne) posety miały identyczną talię kart. Na rysunku 2 widzimy przykład takiej patologicznej pary. Przypuszcza się jednak, że zjawisko to nie występuje wśród posetów o większej liczbie elementów.

*doktorantka, Politechnika Warszawska

Hipoteza 1. *Każdy zbiór częściowo uporządkowany o co najmniej czterech elementach jest rekonstruowalny na podstawie swojej talii kart.*

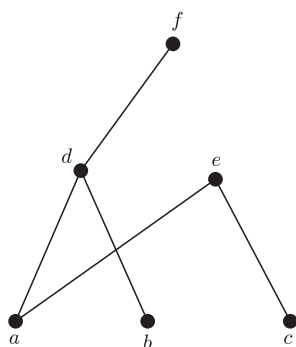
Powyższa hipoteza jest posetową wersją słynnego problemu rekonstrukcji grafów z 1929 r. autorstwa Stanisława Ulama. Pomimo sporych wysiłków pozostaje jak dotąd otwarta.

Przykłady rekonstrukcji

Elementem *minimalnym* w posecie $P = (X, <)$ nazywamy taki element a , że nie zachodzi $x < a$ dla żadnego $x \in X$. Elementem *najmniejszym* posetu P nazywamy taki element b , że $b < x$ dla każdego $x \in X - b$. Widzimy, że element najmniejszy jest zarazem minimalny, ale nie musi być na odwrót. Poset może zawierać wiele elementów minimalnych i wtedy nie ma elementu najmniejszego.

Twierdzenie 1. *Zbiory częściowo uporządkowane z najmniejszym elementem są rekonstruowalne.*

Dowód. Rozważmy talię n -elementowego posetu P ($n \geq 4$). Zauważmy, że jeśli P ma element najmniejszy a , to pojawi się on we wszystkich $n - 1$ kartach posetów P_x , $x \neq a$. Jeśli natomiast P nie ma elementu najmniejszego, to ma co najmniej dwa elementy minimalne, czyli co najwyżej dwie karty P będą miały element najmniejszy. Zatem na podstawie ilości kart z najmniejszym elementem możemy rozpoznać, czy rozważany zbiór ma element najmniejszy, czy go nie ma.



Rys. 3. Na rysunku przedstawiono 6-elementowy zbiór częściowo uporządkowany P . Łańcuchami w P są podzbiory: $\{a, d, f\}$, $\{a, d\}$, $\{a, f\}$, $\{d, f\}$, $\{a, e\}$, $\{b, d, f\}$, $\{b, d\}$, $\{b, f\}$, $\{c, e\}$. Antyłańcuchami w P są podzbiory: $\{a, b, c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{d, e\}$, $\{d, c\}$, $\{e, b\}$, $\{f, e\}$, $\{f, c\}$. Najliczniejszy łańcuch wyznacza wysokość posetu: $h(P) = 2$. Najliczniejszy antyłańcuch wyznacza szerokość posetu: $w(P) = 3$. Elementy a, b, c są minimalne, co oznacza, że zbiór nie ma elementu najmniejszego.

Pozostaje sposób rekonstrukcji. Wiemy już, że P ma element najmniejszy a . Należy odnaleźć kartę, w której usuniętym elementem było właśnie a . Jak to zrobić? Spośród kart wybierzmy taką, która na jak najniższym poziomie diagramu ma najwięcej elementów, ale co najmniej dwa (jeśli takiej nie ma, to porządek P jest liniowy). Do tej karty dokładamy najmniejszy element a i w ten sposób otrzymujemy rekonstrukcję zbioru P . \square

Analogicznie można dowieść rekonstruowalności posetów z największym elementem.

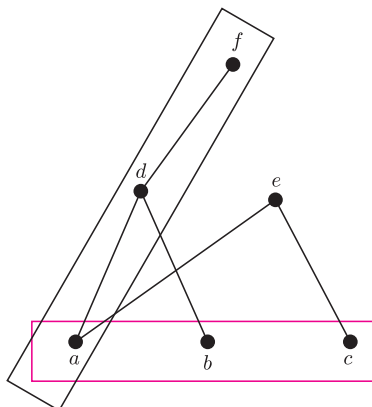
Do elementarnych dowodów należy również pokazanie, że wysokość i szerokość posetu są parametrami rekonstruowalnymi. *Łańcuchem* w posecie P nazywamy zbiór elementów parami porównywalnych, natomiast *antyłańcuchem* nazywamy zbiór, którego żadne dwa elementy nie są porównywalne. *Wysokość* posetu $h(P)$ równa jest liczności najdłuższego łańcucha w nim zawartego minus jeden. *Szerokość* posetu $w(P)$ równa jest liczności najliczniejszego antyłańcucha w nim zawartego.

Twierdzenie 2. *Szerokość i wysokość posetu są rekonstruowalne.*

Dowód. O zbiorze P założmy, że nie jest łańcuchem i nie jest antyłańcuchem. Niech C będzie najliczniejszym łańcuchem w P . Z założenia istnieje w P punkt $x \notin C$, stąd $h(P) = h(P - x)$. A zatem wysokość zbioru P równa jest maksimum wysokości kart zbioru P .

Niech teraz A będzie najliczniejszym antyłańcuchem w P . Ponieważ P nie jest antyłańcuchem, więc istnieje w P punkt $y \notin A$. Zatem A jest najliczniejszym antyłańcuchem również w zbiorze $P - y$, czyli $w(P) = w(P - y)$. Stąd wnioskujemy, że szerokość zbioru P jest równa maksimum szerokości kart zbioru P . \square

Powyższe rozważania na temat rekonstrukcji zbiorów należy uzupełnić o listę innych znanych klas skończonych rekonstruowalnych zbiorów częściowo uporządkowanych. Są to, między innymi, zbiory niespójne, zbiory o niespójnym grafie nieporównywalności, zbiory szerokości 2, łańcuchy i antyłańcuchy. Warto nadmienić także, że istnieją zbiory nieskończone, które nie są rekonstruowalne. Może Czytelnik spróbuje znaleźć przykład?



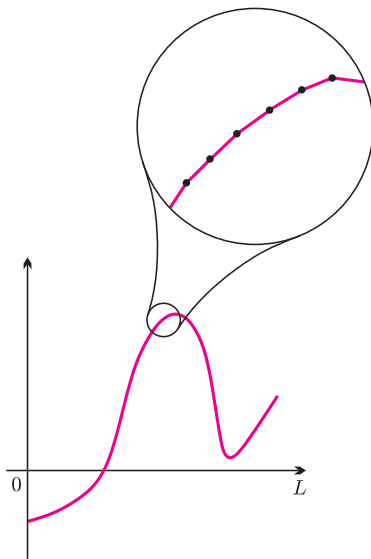


Patrz *Delta* 3 i 6/2009.

Czy komputer potrafi rysować wykresy funkcji?

Szukając kolejnych słabych stron zastosowania komputera w zadaniach matematycznych, obaj bracia zaczęli podejrzewać, że komputer – w zależności od użytego oprogramowania – może mieć kłopoty z narysowaniem wykresów nawet prostych (zdawałoby się) funkcji.

Jak *naprawdę* wygląda wykres funkcji $x \mapsto \sin(kx)$ na odcinku $[0, L]$? Zdawałoby się, że nie ma wątpliwości: im większe k , tym bardziej wije się sinusoida i tym więcej ma miejsc zerowych. Aby się o tym naocznie przekonać, Wojtek skorzystał z darmowego oprogramowania: Maximy oraz Octave, jego brat mógł na uczelni przetestować komercyjne pakiety Mathematica oraz MATLAB. Rzeczywiście, dla $k = 1, 2, 3$ oczekiwania potwierdziły się i Wojtek ujrzał doskonale znane mu wykresy. Oprogramowanie zadziałało bez zarzutu, jednak w głowie Wojtka kiełkował już szatański pomysł: „Czy nie dałoby się przypadkiem dobrać takiego $k > 1$, dla którego wykres $\sin(kx)$ zostanie pokazany tak, jakby była to funkcja... $\sin(x)$?!”



Wojtek wiedział bowiem, że programy komputerowe kreślą wykresy funkcji, wyznaczając ich wartości w bardzo wielu punktach i następnie łącząc te wartości łamaną. Ponieważ miejsc próbkowania funkcji jest bardzo dużo, mamy złudzenie, że łamana jest gładka, co dobrze widać na ilustracji.

Wojtek postanowił wykorzystać właśnie tę słabość: przecież przez zadane punkty na płaszczyźnie można poprowadzić nieskończenie wiele krzywych! „Skoro program łączy te punkty łamaną, to wystarczy wybrać taką funkcję, która *w tych punktach* przyjmuje te same wartości co $\sin(x)$...” Wszystko fajnie, tylko pozostawał jeden problem: w *których konkretnie* punktach program będzie sprawdzać wartości funkcji? Wojtek postanowił zaryzykować i przyjął, że program idzie po linii najmniejszego oporu i próbuje funkcję w $N + 1$ równo odległych węzłach. Na odcinku $[0, L]$ byłyby więc to punkty

$$x_i = ih, \quad \text{gdzie } h = L/N, i = 0, \dots, N.$$

Wtedy, żeby zachodziło $\sin(x_i) = \sin(kx_i)$, wystarczyłoby, że dla pewnego całkowitego m

$$kx_i = x_i + 2m\pi \quad \text{dla } i = 0, \dots, N,$$

skąd od razu widać, że jednym z dobrych wyborów jest $k = 1 + 8\pi/h$. Jednak dalej nie wiemy przecież, jakie jest N , a tym samym h ... Wojtek niezrażony brnął jednak dalej: w końcu matematyk przy pracy musi być optymistą! Wciąż licząc na to, że program implementuje rozwiązania najprostsze z możliwych, założył, że N jest parzyste. „Na pewno uwzględniają wartości funkcji w krańcach odcinka i do nich dokładają jeszcze parę wartości ze środka przedziału $(0, L)$. Zapewne wszystkie inne punkty tworzone są przez dwukrotne zagęszczenie tego wyjściowego podziału – a to znaczyłoby, że N jest całkowitą wielokrotnością jakiejś potęgi dwójki! Zresztą, sprawdźmy w dokumentacji...”

Sięgnął do programu Octave, którego funkcja `fplot` rzekomo potrafi rysować wykresy nawet bardzo skomplikowanych funkcji. Ponieważ jest to wolne oprogramowanie, Wojtek miał dostęp do pełnego kodu źródłowego. Okazało się, że wszystkie jego przypuszczenia potwierdziły się co do joty, a procedura zagęszczania startowała z podziału odcinka na 7 części. Skoro tak, to należy wziąć $k = 1 + 7\pi 2^n / L$ dla dostatecznie dużego n . „Na przykład, gdy $L = 5$, to dla $n = 4$ dostajemy $k \approx 71,372$. Wpiszmy zatem do Octave:”



```
fplot("sin(71.372*x)",[0,5]);
```

Efekt przeszedł jego najśmielsze oczekiwania: jego oczom ukazał się, zamiast szybkozmiennej funkcji, wykres... $\sin(x)$. „Hurraaa!!!...”

W ciągu następnych paru minut metodą prób i błędów wykrył, że w Maximie ten sam efekt na odcinku $[0, 5]$ uzyskuje się dla funkcji $\sin(403,124x)$: zupełnie jakby Maxima „myślała”, że wykresem tej funkcji jest wykres $\sin(x)$! Rzut oka w dokumentację MuPADA sprawił, że chwilę później ten system „poległ” na $\sin(151,8x)$. Natomiast dla Matematiki Wojtkowi nie udało się (jeszcze) znaleźć odpowiedniego przykładu...

„Cóż, mylić się jest rzeczą *nie tylko* ludzką...” filozoficznie skwitował Wojtek. Ziewnął, przeciągnął się – i poszedł grać w kosza.

Czy mimo to komputer może pomóc matematykowi?

Przekonanie o tym, że komputery są niezawodne i znacznie sprawniejsze w rachunkach niż człowiek, jest powszechne. Przekonanie to prawdopodobnie w jakimś stopniu podziela nawet Ministerstwo Nauki i Szkolnictwa Wyższego, zalecając w możliwie dużym stopniu wykorzystywanie pakietów komputerowych do wykładania matematyki na uczelniach. Należy wszakże pamiętać, że komputer to tylko narzędzie o ograniczonych możliwościach. Możemy z jego pomocą *eksperymentować* z matematyką, pamiętając, że wyniki każdego eksperymentu są podatne na zaburzenia wprowadzane przez „niedokładności lub wady sprzętu laboratoryjnego”, a także „nieuwagę lub niedbałość laboranta”. Użycie komputera – zwłaszcza w matematyce – nie zwalnia od myślenia; przeciwnie, wymaga głębszej refleksji, wiedzy, spostrzegawczości i kultury matematycznej! Na szczęście komputer nie jest nieprzewidywalną wróżką. To potężne narzędzie, które w umiejętnych rękach może zdziałać cuda – lecz w dłoniach osoby niemającej dostatecznej wiedzy o tajnikach jego działania może być bardziej niebezpieczne od brzytwy...

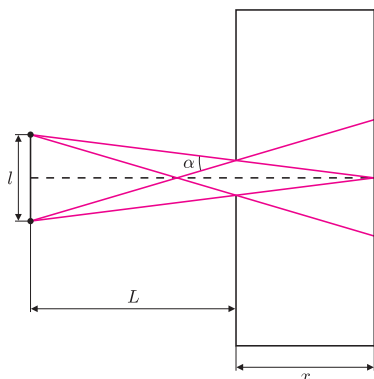
Konkurs! Zapraszamy Czytelników do wskazania innych ciekawych przykładów nieprawidłowego rysowania wykresów funkcji w pakietach komputerowych. Przykłady mogą dotyczyć działania jednego z trzech systemów: Maxima, Mathematica i MuPAD. Przykład powinien zawierać: definicję (tzn. wzór) funkcji, nazwę wykorzystanego oprogramowania, polecenie systemowe użyte do narysowania wykresu oraz opis, na czym polega różnica między prawdziwym wykresem a tym, co narysuje system. Propozycje prosimy nadsyłać wyłącznie pod adresem e-mail knn@mimuw.edu.pl, z tematem: „Konkurs rysunkowy”. **Autorów najciekawszych propozycji nagrodzimy publikacją** ich przykładów w *Delcie*.

Małą Deltę przygotował Piotr KRZYŻANOWSKI



Rozwiązanie zadania F 749.

Camera obscura to po prostu skrzynka z dziurką, naprzeciw której znajduje się klisza lub papier światłoczuły (konstrukcja taka jest nadal używana współcześnie do tzw. *fotografii otworkowej*). Aby można było rozróżnić pręgi, odległość kliszy od otwórka powinna być taka, żeby obrazy pręgi nie pokrywały się ani nie stykały.



Z rozważań geometrycznych wynika, że:

$$x > \frac{d}{2} \operatorname{ctg} \alpha.$$

Ale

$$\operatorname{tg} \alpha = \frac{l}{2(L+x)}, \text{ zatem } x > \frac{d(L+x)}{l}$$

i stąd:

$$x > \frac{dL}{l} = 10 \text{ cm.}$$

Zatem głębokość skrzynki powinna być większa niż 10 cm.

Czy naprawdę *prawie* robi wielką różnicę?

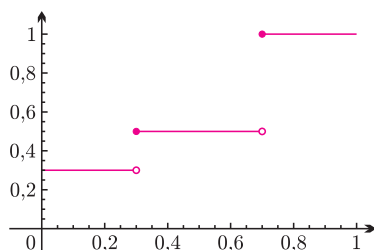
Paulina MAŁOLEPSZA, Tomasz MAŁOLEPSZY*



Jednym z fundamentalnych pojęć analizy matematycznej jest bez wątpienia różniczkowalność. Dla funkcji jednej zmiennej, określonej na pewnym otwartym przedziale, równoważna jest ona istnieniu pochodnej funkcji w każdym punkcie tego przedziału. Jak wiadomo, wszystkie funkcje elementarne są różniczkowalne w tym klasycznym sensie, jednak wiele innych prostych i zarazem użytecznych funkcji już nie. Przykładem jest chociażby funkcja $y = |x|$, dla której wszystko psuje się w zerze, w którym pochodna nie istnieje. Okazuje się jednak, że można tak osłabić pojęcie różniczkowalności, że wyżej wspomniana funkcja będzie już różniczkowalna w tym nowym sensie. Wystarczy w tym celu wziąć pod uwagę tzw. różniczkowalność prawie wszędzie. Cóż to oznacza? Pojęcie *prawie wszędzie* (będziemy też pisać p.w.) jest krótszym określeniem na sformułowanie „wszędzie poza zbiorem (jednowymiarowej) miary Lebesgue’a zero”. Wymyślona na początku XX wieku przez Henri Lebesgue’a koncepcja miary, nazwanej później jego nazwiskiem, jest niczym innym jak uogólnieniem pojęcia długości przedziału. Dzięki tej mierze można jednak mierzyć dodatkowo „długości” wielu innych zbiorów zawartych w prostej. Można np. zmierzyć „długość” punktu (jego miara to zero), zbioru liczb naturalnych w \mathbb{R} (miara tego zbioru jest również równa zero) lub „długość” zbioru liczb niewymiernych w $[0, 1]$ (miara w tym przypadku wynosi jeden). Intuicyjnie czujemy, że jeżeli miara Lebesgue’a jakiegoś zbioru wynosi 0, to musi on być „bardzo mały”. I choć dopuszczenie do sytuacji, że funkcja nie musi być różniczkowalna na takim „małym” zbiorze, wpłynie z pewnością na jej zachowanie i własności, to czy skutkiem tego może być radykalna zmiana faktów znanych nam z klasycznej analizy matematycznej?

Aby odpowiedzieć na to pytanie, na początku odnotujmy następujące twierdzenie.

Twierdzenie 1. *Jedyną funkcją f ciągłą w przedziale $[0, 1]$ oraz spełniającą wszędzie w przedziale $(0, 1)$ warunek $f'(x) = 0$ jest funkcja stała.*



Rys. 1

Czy to twierdzenie pozostanie prawdziwe, gdy w miejsce słowa „wszędzie” wstawimy „prawie wszędzie”? Na pierwszy rzut oka wydaje się, że tak. Oczywiście, można z łatwością podać (patrz rysunek 1) przykład niemalejącej funkcji różniczkowalnej prawie wszędzie o pochodnej równej zero, która nie jest stała, ale tę monotoniczność uzyskaliśmy za cenę nieciągłości funkcji. Czy jednak mimo wszystko mogą istnieć różne od stałej **CIĄGŁE** funkcje różniczkowalne p.w. o pochodnej równej zero, które są, powiedzmy, niemalejące? I choć intuicja podpowiada, że takich obiektów matematycznych nie ma, to jak powiedział brytyjski matematyk Edward Titchmarsh „być może najbardziej zaskakujące w matematyce jest to, że jest tak zaskakująca”. Tak, są takie funkcje! Jedną z nich jest funkcja Cantora (zwaną też funkcją Cantora–Lebesgue’a), pochodząca z 1883 roku. Do jej zdefiniowania będzie nam potrzebny tzw. zbiór Cantora. Przypomnijmy jego konstrukcję.

Zaczynamy od odcinka $[0, 1]$. Dzielimy go na trzy podprzedziały równej długości i środkowy z nich, otwarty, oznaczamy przez I_1^1 . W kolejnym kroku pozostałe dwa przedziały również dzielimy na trzy podprzedziały każdy i środkowe z nich, otwarte, oznaczamy przez I_2^1 oraz I_2^2 . W kolejnych krokach powtarzamy tę procedurę, tzn. po n -tym kroku otrzymujemy 2^{n-1} przedziałów otwartych I_n^i , $i = 1, \dots, 2^{n-1}$, o długości $\frac{1}{3^n}$ każdy. Zbiór Cantora określamy wówczas jako zbiór C postaci

$$C = [0, 1] \setminus G,$$

gdzie

$$G = \bigcup_{n=1}^{\infty} \bigcup_{i=1}^{2^{n-1}} I_n^i.$$

Zbiór Cantora ma intrygujące własności. Przede wszystkim jest przykładem domkniętego zbioru nieprzeliczalnego miary Lebesgue’a zero. Co więcej, należą do niego tylko i wyłącznie te liczby z przedziału $[0, 1]$, które da się przedstawić w systemie trójkowym bez użycia 1.

*Wydział Matematyki, Informatyki i Ekonometrii, Uniwersytet Zielonogórski

Oznaczmy teraz

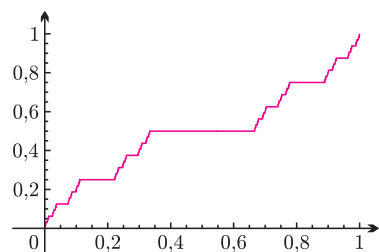
$$G_k = \bigcup_{n=1}^k \bigcup_{i=1}^{2^{n-1}} I_n^i.$$

Dla dowolnego naturalnego k zdefiniujmy funkcję $f_k : [0, 1] \rightarrow [0, 1]$ następująco:

$$f_k(0) = 0, \quad f_k(1) = 1,$$

$$f_k(x) = \frac{2i-1}{2^n} \quad \text{dla } x \in \overline{I_n^i}, \quad \text{gdzie } n = 1, \dots, k, \quad i = 1, \dots, 2^{n-1}$$

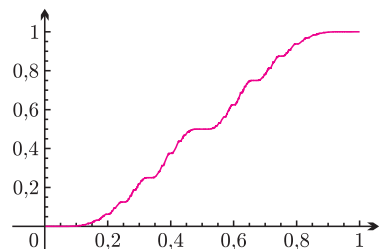
($\overline{I_n^i}$ oznacza domknięcie zbioru I_n^i), natomiast w domknięciach przedziałów dopełniających zbiór G_k do przedziału $[0, 1]$ okreśmy funkcję f_k jako ciągłą i liniową, tak aby w całym przedziale $[0, 1]$ funkcja f_k była ciągła. Funkcja f_k będzie więc niemalejąca i ciągła w $[0, 1]$. Co istotne, można wykazać, że ciąg funkcyjny $\{f_k\}_{k=1}^\infty$ jest jednostajnie zbieżny, a więc jego granica f_C istnieje i jest funkcją ciągłą. Granicę tę nazywamy właśnie funkcją Cantora. Jest ona niemalejąca (bo wszystkie f_k są niemalejące) oraz stała w każdym z przedziałów I_n^i . Stąd wynika od razu, że poza zbiorem Cantora funkcja f_C ma pochodną równą 0. Widzimy więc, że istnieje funkcja ciągła, która jest określona na przedziale długości 1 i której pochodna na zbiorze miary 1 jest równa 0, a mimo to nie jest funkcją stałą. Co więcej, przyjmuje wszystkie wartości pomiędzy 0 i 1! Zaiste, dziwna jest ta funkcja. Zresztą czasami nazywa się ją również, ze względu na nieskończoną ilość „schodków” (czyli tych fragmentów jej wykresu, na których jest stała), „diabelskimi schodami”, co także podkreśla jej nieoczekiwane własności. Z drugiej strony jednak, skoro istnieje taka funkcja jak funkcja Cantora, to może warto zapytać, czy istnieje określona w przedziale $[0, 1]$ ŚCIŚLE monotoniczna funkcja CIĄGŁA, która prawie wszędzie w $(0, 1)$ będzie miała pochodną równą 0? I tu, co zaskakujące, odpowiedź jest twierdząca! W literaturze znanych jest wiele funkcji o tej własności (np. funkcje Riesz–Nagy’a), jednak tutaj podamy przykład chyba najslawniejszej wśród nich, a mianowicie funkcji Minkowskiego $?(x)$, która po raz pierwszy na kartach historii matematyki pojawia się w 1904 roku. Można ją określić na wiele sposobów. My do tego celu wykorzystamy ułamki łańcuchowe. Przypomnijmy, że każdą liczbę rzeczywistą x można przedstawić w postaci



Rys. 2. Funkcja Cantora.

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots}}},$$

(przedstawienie to w skrócie można zapisać również jako $x = [a_0; a_1, a_2, \dots]$), gdzie $a_0 \in \mathbb{Z}$, $a_i \in \mathbb{N}$ dla $i = 1, 2, \dots$. Co istotne, każda liczba niewymierna ma dokładnie jedno takie przedstawienie (w postaci nieskończonego ułamka łańcuchowego), dowolną zaś liczbę wymierną można zapisać w dwóch postaciach: krótszej $[a_0; a_1, a_2, \dots, a_n]$, gdzie $a_n \neq 1$, lub dłuższej $[a_0; a_1, a_2, \dots, a_n - 1, 1]$. Oznaczmy $R_k = \sum_{j=1}^k a_j$. Wówczas na przedziale $[0, 1]$ funkcja Minkowskiego (zwana po angielsku *question mark*) zdefiniowana jest następująco:



Rys. 3. Funkcja Minkowskiego.

$$?(x) = \begin{cases} 2 \sum_{i=1}^n \frac{(-1)^{i-1}}{2^{R_i}} & \text{dla } x \text{ wymiernych oprócz } 1, \\ 2 \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{2^{R_i}} & \text{dla } x \text{ niewymiernych,} \\ 1 & \text{dla } x = 1. \end{cases}$$

Wykazanie, że $?(x)$ ma wymagane przez nas własności, czyli ciągłość, ścisłą monotoniczność, pochodną prawie wszędzie równą 0, jest o wiele trudniejsze niż w przypadku funkcji Cantora. Niemniej jednak fakt istnienia takiej funkcji pozwala nam sformułować następujące twierdzenie.

Twierdzenie 2. *Istnieją ściśle monotoniczne funkcje ciągłe f określone w przedziale $[0, 1]$ oraz spełniające prawie wszędzie w przedziale $(0, 1)$ warunek $f'(x) = 0$.*

Jak widać, zmiana jakościowa między twierdzeniami 1 i 2 jest znaczna. Zatem także w matematyce *prawie* potrafi zrobić wielką różnicę. . .

**Rozwiązanie zadania M 1255.**

Niech n będzie najmniejszą liczbą, dla której teza zadania nie jest spełniona. Przyjmijmy, że wówczas $a < 2^{n-1}$. Ponieważ liczby a, b, c, d są dodatnie, więc $n > 1$. Wobec tego prawa, a zatem i lewa strona danej równości jest podzielna przez 8. Ponieważ kwadrat liczby całkowitej daje z dzielenia przez 8 resztę 0, 1 lub 4, więc liczby a, b, c, d , spełniające daną zależność, muszą być parzyste, tzn. $a = 2a_1, b = 2b_1, c = 2c_1, d = 2d_1$. Stąd wniosek, że $a_1 < 2^{n-2}$ oraz

$$a_1^2 + b_1^2 + c_1^2 + d_1^2 = 7 \cdot 4^{n-1},$$

co oznacza, że teza zadania nie jest spełniona również dla liczby $n - 1$. Otrzymaliśmy sprzeczność.

**Rozwiązanie zadania M 1256.**

Przypuśćmy, że $a_1, a_2, \dots, a_n < 2$ oraz niech $b_i = 2 - a_i$ ($i = 1, 2, \dots, n$). Wówczas $b_i > 0$ oraz

$$b_1 + b_2 + \dots + b_n \leq n.$$

Ponadto

$$\begin{aligned} n^2 &\leq \sum_{i=1}^n (2 - b_i)^2 = \\ &= 4n - 4 \sum_{i=1}^n b_i + \sum_{i=1}^n b_i^2 < \\ &< 4n - 4 \sum_{i=1}^n b_i + \left(\sum_{i=1}^n b_i \right)^2. \end{aligned}$$

Niech $B = b_1 + b_2 + \dots + b_n$. Wówczas $n(n - 4) < B(B - 4)$. Wiemy, że $n \geq 4$, więc liczba $B(B - 4)$ jest dodatnia. Ponadto $B > 0$, więc $B - 4 > 0$.

Wiemy też, że $B \leq n$, więc $B - 4 > n - 4$, czyli $B > n$. Otrzymaliśmy sprzeczność.

Informatyczny kącik olimpijski (24): Barykady

W finale Potyczek Algorytmicznych 2007 zawodnicy mieli do rozwiązania, między innymi, zadanie *Barykady*. Mając dane drzewo (czyli spójny graf nieskierowany o n wierzchołkach i $n - 1$ krawędziach), dla każdej z podanych liczb dodatnich k_i należało wyznaczyć minimalną liczbę krawędzi, które należy z tego drzewa usunąć, aby jedna z pozostałych po tym zabiegu spójnych części zawierała dokładnie k_i wierzchołków.

Rozwiązanie opiera się na programowaniu dynamicznym. Wybierzmy najpierw dowolny wierzchołek i uznajmy go za korzeń drzewa. Będziemy chcieli dla każdego poddrzewa wyznaczonego przez wierzchołek (czyli fragmentu złożonego z tego wierzchołka i wszystkich jego potomków) i każdej liczby całkowitej dodatniej x wyznaczyć minimalną liczbę krawędzi, taką że po usunięciu z tego poddrzewa tych krawędzi w poddrzewie zostaje spójny fragment zawierający dokładnie x wierzchołków, z których jednym jest korzeń tego poddrzewa.

Dla liści jest to bardzo łatwe: można to zrobić tylko dla $x = 1$, nie usuwając z poddrzewa (złożonego przecież tylko z samego liścia) żadnych krawędzi. Aby obliczyć wyniki dla pewnego węzła wewnętrznego v i wszystkich możliwych wartości x , należy najpierw uczynić to dla wszystkich jego synów v_1, v_2, \dots, v_m . Niech w_{v_i} oznacza tablicę wyników dla poddrzewa wierzchołka v_i , a s_i – tablicę dla wierzchołka v wraz z poddrzewami v_1, v_2, \dots, v_i . Wynikiem dla v jest wówczas tablica s_m . Łatwo zauważyć, że $s_1[x + 1] = w_{v_1}[x]$ (gdyż przecinając te same krawędzie, mamy poddrzewo z jednym wierzchołkiem więcej) oraz $s_1[1] = 1$ (przecinamy krawędź łączącą v z v_1). Ponadto, dla $i \geq 2$:

$$(*) \quad s_i[x] = \min\left(\min_{j=1,2,\dots,x-1} (s_{i-1}[j] + w_{v_i}[x - j]), s_{i-1}[x] + 1\right).$$

Powyższa równość wynika z tego, że możemy z poddrzewa wierzchołka v_i wziąć pewną liczbę wierzchołków (konkretnie: $x - j$) i dołączyć je do stworzonego wcześniej poddrzewa (o j wierzchołkach) albo nie wziąć żadnych wierzchołków z tego poddrzewa, tj. przeciąć krawędź łączącą v z v_i .

Obliczywszy tablice wyników w_v dla wszystkich wierzchołków, otrzymujemy wynik dla k_i równy (r jest korzeniem drzewa):

$$\min(w_r[k_i], \min_{v \in V \setminus \{r\}} (w_v[k_i] + 1)).$$

W powyższym napisie dodatkowa jedynka w przypadku $v \neq r$ wzięła się stąd, że musimy przeciąć krawędź łączącą v z jego ojcem, czego nie uwzględnialiśmy przy liczeniu w_v .

Pozostaje jeszcze zastanowić się, jak szybko działa algorytm oparty na opisanych pomysłach. Wzór (*) na elementy s_i daje bezpośrednią metodę liczenia tablic w_v . Widać, że algorytm ten ma złożoność $O(n^3)$, gdyż każde poddrzewo dołączamy raz, w czasie $O(n^2)$, do jakiegoś obliczonego wcześniej s_i . Udowodnimy teraz, że algorytm ten działa istotnie szybciej. Wielu finalistów Potyczek nie dostrzegło tego faktu i w związku z tym nawet nie próbowało implementować omawianego rozwiązania, myśląc, że będzie zbyt wolne. Rzeczoną lepiej oszacowaną złożonością algorytmu jest $O(n^2)$.

Oznaczmy przez $a(v)$ rozmiar poddrzewa o korzeniu w v . Udowodnię indukcyjnie po $a(v)$, że łączna liczba operacji wykonywanych w poddrzewie o korzeniu w v jest $O(a(v)^2)$. Dla liści ($a(v) = 1$) jest to niewątpliwie zdanie prawdziwe. Weźmy teraz węzeł v i jego synów v_1, v_2, \dots, v_m . W algorytmie najpierw obliczamy tablice wynikowe dla synów, co daje nam $O(a(v_1)^2 + a(v_2)^2 + \dots + a(v_m)^2)$ operacji, zgodnie z założeniem indukcyjnym. Zauważmy, że w tablicy s_{i-1} jest $1 + a(v_1) + a(v_2) + \dots + a(v_{i-1})$ elementów, a w tablicy w_i jest $a(v_i)$ elementów. Stąd utworzenie tablicy s_i z tablic s_{i-1} i w_i zgodnie ze wzorem (*) zajmuje czas

$$O(\text{rozmiar}(s_{i-1}) \cdot \text{rozmiar}(w_i)) = O((1 + a(v_1) + a(v_2) + \dots + a(v_{i-1})) \cdot a(v_i))$$

(dlaczego?). W takim razie wszystkie obliczenia w poddrzewie wierzchołka v zajmują czas

$$\begin{aligned} O\left(a(v_1)^2 + \dots + a(v_m)^2 + \sum_{1 \leq i < j \leq m} a(v_i)a(v_j) + \sum_{i=1}^m a(v_i)\right) &= O\left(\left(\sum_{i=1}^m a(v_i) + 1\right)^2\right) \\ &= O(a(v)^2), \end{aligned}$$

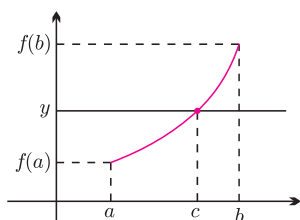
co dowodzi, że w całym drzewie wykonamy zaledwie $O(n^2)$ operacji.

Tomasz KULCZYŃSKI

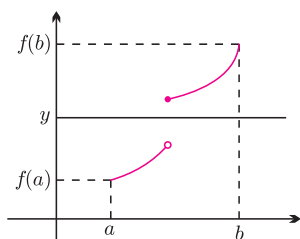
O własności Darboux

Witold BEDNAREK*

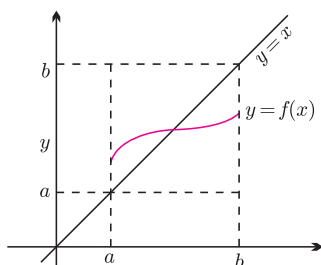
Zobacz artykuł Jerzego Mioduszewskiego „Bernard Bolzano (1781–1848), uczonec i matematyk” w *Matematyce* 8/2007, str. 451–455.



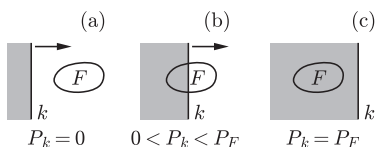
Rys. 1



Rys. 2



Rys. 3



Rys. 4

Znacznie trudniej jest wykazać, że na płaszczyźnie istnieje prosta dzieląca jednocześnie na połowy pola dwóch figur. Natomiast dla trzech figur taka prosta zazwyczaj nie istnieje. Prosty kontrprzykład to trzy koła o niewspółliniowych środkach.

Na wstępie odnotujmy, że Jean Gaston Darboux (1842–1917), matematyk francuski zajmujący się przede wszystkim geometrią różniczkową, nie był odkrywcą tytułowej własności. Twierdzenie, które będziemy rozważać, sformułował i udowodnił czeski matematyk Bernard Bolzano (1781–1848), niedoceniony za życia – podstawowa jego praca, *Nauka o funkcjach*, została wydana dopiero w 1930 r.

Twierdzenie 1 (Własność Darboux). Niech $f : [a, b] \rightarrow \mathbb{R}$ będzie funkcją ciągłą. Wówczas dla dowolnego y znajdującego się pomiędzy $f(a)$ i $f(b)$, czyli $f(a) \leq y \leq f(b)$ lub $f(b) \leq y \leq f(a)$, istnieje takie $c \in (a, b)$, że $f(c) = y$.

Rysunek 1 ilustruje powyższe twierdzenie. Dla funkcji nieciągłej twierdzenie 1 może być fałszywe (zob. rys. 2).

Twierdzenia 1 nie można odwrócić. Przykładem jest funkcja $f : [-1, 1] \rightarrow [-1, 1]$ określona wzorem

$$\begin{cases} \sin \frac{1}{x} & \text{dla } x \neq 0, \\ 0 & \text{dla } x = 0. \end{cases}$$

Czytelnik Wnikliwy sprawdzi, że funkcja f ma własność Darboux, ale nie jest ciągła w punkcie $x = 0$. Ale pod pewnymi dodatkowymi warunkami z własności Darboux wynika ciągłość funkcji.

Twierdzenie 2. Jeśli funkcja jest ściśle monotoniczna w przedziale domkniętym i ma własność Darboux, to jest ciągła w tym przedziale.

Własność Darboux najczęściej znajduje zastosowanie przy badaniu istnienia i liczby miejsc zerowych funkcji ciągłych. Jeśli bowiem $f : [a, b] \rightarrow \mathbb{R}$ jest ciągła oraz $f(a) \cdot f(b) < 0$, to $f(c) = 0$ dla pewnego $c \in (a, b)$. Nierówność $f(a) \cdot f(b) < 0$ to krótszy sposób zapisania alternatywy $f(a) < 0$ i $f(b) > 0$ lub $f(a) > 0$ i $f(b) < 0$. Na przykład równanie $x^4 = x + 1$ ma rozwiązanie w przedziale $(-1, 0)$, gdyż funkcja ciągła $f(x) = x^4 - x - 1$ spełnia $f(-1) = 1 > 0$ oraz $f(0) = -1 < 0$.

Przyjmijmy teraz, że funkcja ciągła $f : [a, b] \rightarrow [a, b]$ spełnia warunki $f(a) > a$ i $f(b) < b$. Na rysunku 3 wykres funkcji f przecina prostą o równaniu $y = x$, czyli wykres funkcji identycznościowej. Istnienie punktu przecięcia tych dwóch wykresów oznacza, że równanie $f(x) = x$ ma rozwiązanie. Okazuje się, że tak jest dla dowolnej funkcji f spełniającej powyższe warunki. Oto łatwe uzasadnienie tego spostrzeżenia. Niech $g(x) = f(x) - x$ dla $x \in [a, b]$. Funkcja g , oczywiście, jest ciągła, a ponadto $g(a) = f(a) - a > a - a = 0$ i $g(b) = f(b) - b < b - b = 0$. Zatem istnieje taki punkt $c \in (a, b)$, że $g(c) = 0$, czyli $f(c) = c$. Punkt c to punkt stały funkcji f .

Powyższe spostrzeżenie ma dalekosiężne uogólnienie – twierdzenie Brouwera o punkcie stałym.

Twierdzenie 3. Niech B będzie n -wymiarową kulą domkniętą w przestrzeni \mathbb{R}^n oraz $T : B \rightarrow B$ będzie odwzorowaniem ciągłym. Wówczas istnieje taki punkt $c \in B$, że $T(c) = c$ (punkt stały odwzorowania T).

Okazuje się, że własność Darboux można zastosować również w geometrii elementarnej. Oto przykład: każdą figurę płaską o niepustym wnętrzu, domkniętą i ograniczoną, można podzielić prostą na dwie części o równych polach. Niech bowiem P_F oznacza pole figury F , P_k zaś oznacza pole części wspólnej figury F i półpłaszczyzny o krawędzi k (po lewej stronie prostej k). Przesuwając prostą k od lewej do prawej (zob. rys. 4), mamy na początku $P_k = 0$, a na końcu $P_k = P_F$. Pole P_k jest ciągłą funkcją długości wektora przesunięcia, więc dla pewnej prostej k_0 zachodzi równość $P_{k_0} = \frac{1}{2} P_F$.

Powyższe stwierdzenie można przenieść na wyższe wymiary. Dla wymiaru 3 jest to znane twierdzenie o podziale kanapki.

Twierdzenie 3. Jeśli na kromce chleba umieścimy w dowolny sposób masło i ser, to można jednym (płaskim) cięciem przepolować wszystkie trzy składniki kanapki (tzn. ich objętość).

Dla ogólnej kanapki, zawierającej przynajmniej cztery składniki, twierdzenie nie jest prawdziwe. Być może dlatego matematycy nie przypisują większego znaczenia urom sztuki kulinarnej.

*nauczyciel liceum i gimnazjum w Łodzi

Wyznaczamy moduł Younga

Stanisław BEDNAREK

Pod działaniem sił zewnętrznych wszystkie ciała ulegają odkształceniom. Jednym z rodzajów odkształceń jest wydłużenie, czyli zwiększenie długości ciała (rys. 1). Wielkość tego wydłużenia, Δl , zależy od wartości działającej siły F , pola przekroju poprzecznego S ciała, jego długości początkowej l_0 oraz właściwości sprężystych. Właściwości te charakteryzuje moduł Younga oznaczany literą E . Przeprowadzone doświadczenia wykazały, że wydłużenie ciała opisuje prawo Hooke'a, które wyraża się następującym wzorem:

$$(1) \quad \Delta l = \frac{Fl_0}{ES}$$

Przekształcając wzór (1) do postaci

$$(2) \quad E = \frac{(F/S)}{(\Delta l/l_0)},$$

można łatwo zauważyć, że moduł Younga równy jest stosunkowi naprężenia, mierzonego stosunkiem siły F do pola przekroju poprzecznego S , do spowodowanego tym naprężeniem wydłużenia względnego $\Delta l/l_0$. Nasze dzisiejsze zadanie będzie polegało na doświadczalnym wyznaczeniu modułu Younga.

W tym celu potrzebne będą: szeroka gumka aptekarska, tzw. recepturka, duży (ok. 0,5 l pojemności) plastikowy kubek od napojów lub od surówki, igła, mocna nić, linijka z podziałką milimetrową, naczynie z wodą, duża szklanka lub inne naczynie w kształcie walca oraz statyw z poprzeczką. Zamiast statywu można także użyć długiego kija, np. od szczotki, położonego na dwóch krzesłach.

Gumkę aptekarską nakładamy na poprzeczkę statywu (rys. 2). Za pomocą igły wykonujemy w górnej części kubka dwa otworki i przewlekamy przez nie nić, tak żeby przechodziła wzdłuż średnicy kubka. Odcinamy kawałek nici o długości około 20 cm i wolne jego końce przekładamy przez gumkę, a następnie zawiązujemy supeł. W ten sposób zawieszamy kubek na gumce aptekarskiej za pomocą nici. Gumka powinna wyprostować się pod ciężarem kubka.

Korzystając z linijki, mierzymy długość l_0 podwójnie złożonej, nieobciążonej gumki oraz jej grubość i szerokość. Odczytane wyniki pomiarów wyrażone są w milimetrach, ale wygodnie będzie przeliczyć je na metry. Przekrój poprzeczny gumki ma kształt prostokąta o bokach a, b . Mnożąc te długości boków i podwajając uzyskany wynik, otrzymujemy przekrój poprzeczny podwójnie złożonej gumki $S = 2ab$, występujący we wzorze (1).

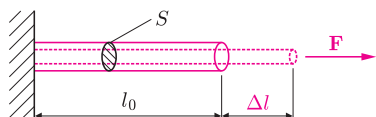
Napełnienie wodą kubka zawieszzonego na nici wywrze na gumkę siłę rozciągającą. Po wleciu wody do kubka mierzymy długość l rozciągniętej gumki. Zmiana długości gumki, Δl we wzorze (1), jest, oczywiście, równa $l - l_0$. Ponieważ ciężar wody wielokrotnie przewyższa ciężar kubka, ten ostatni możemy pominąć w naszych rozważaniach. Siła F , rozciągająca gumkę w naszym układzie doświadczalnym, równa jest ciężarowi wody zawartej w kubku i wyraża się wzorem

$$(3) \quad F = V\rho g,$$

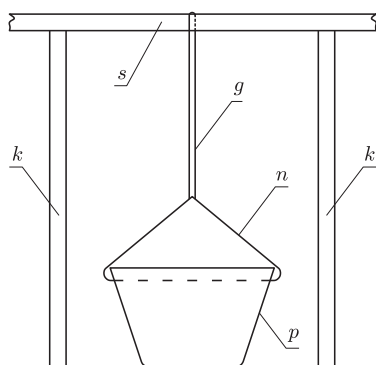
w którym ρ oznacza gęstość wody ($\rho = 1000 \text{ kg/m}^3$), natomiast g jest przyspieszeniem ziemskim ($g = 9,81 \text{ m/s}^2$). Objętość wody V możemy zmierzyć, przelewając ją z kubka do szklanki, mierząc linijką średnicę d szklanki oraz wysokość h słupa wody w szklance i korzystając ze wzoru

$$(4) \quad V = \frac{\pi d^2 h}{4}.$$

Jeżeli szklanka jest mała, możemy wlewać do niej wodę w kilku porcjach i zsumować uzyskane objętości poszczególnych porcji. Wysokość słupa wody w szklance wyznaczamy, zanurzając w niej pionowo linijkę. Na wielu linijkach podziałka milimetrowa nie zaczyna się od jej brzegu – żeby to uwzględnić, kładziemy linijkę na kartce papieru i rysujemy odcinek o długości równej odległości początku skali od początku linijki, a następnie mierzymy długość



Rys. 1. Wydłużenie pręta Δl spowodowane przez siłę F ; l_0 – długość początkowa, S – przekrój poprzeczny.



Rys. 2. Budowa układu doświadczalnego; k – oparcia krzesel, s – kij od szczotki, g – gumka aptekarska, n – nić, p – plastikowy kubek.

tego odcinka w milimetrach. Zwykle długość ta wynosi około 3–4 mm i dodajemy ją do głębokości wody w kubku zmierzonej przez zanurzenie linijki. Opisany wyżej pomiar objętości wody upraszcza się znacznie, jeśli mamy dostęp do naczynia miarowego z podziałką, w które często wyposażona jest domowa kuchnia.

Teraz możemy już obliczać moduł Younga E . W tym celu wzór (1) przekształcamy do postaci

$$(5) \quad E = \frac{Fl_0}{S(l - l_0)}.$$

Do otrzymanego wyrażenia (5) podstawiamy wzór (3), a następnie wzór (4). Otrzymujemy wówczas końcowy wzór

$$(6) \quad E = \frac{\pi g \rho d^2 h l_0}{8ab(l - l_0)},$$

który pozwoli nam obliczyć moduł Younga E po podstawieniu wyników przeprowadzonych pomiarów.

Wartość obliczoną ze wzoru (6) porównujemy z wartością modułu Younga dla gumy podaną w tablicach fizycznych. Wartość tablicowa wynosi $5 \cdot 10^6 \text{ N/m}^2$. Wyznaczona przez nas wartość może różnić się dość znacznie od wartości tablicowej. Przyczyną tej różnicy mogą być zarówno niedokładności wykonanych przez nas pomiarów, jak i fakt istnienia wielu rodzajów gumy różniących się właściwościami sprężystymi.

Na zakończenie problem do samodzielnego rozwiązania. Należy ocenić, jak duży błąd wyznaczenia modułu Younga spowodowany jest niedokładnościami poszczególnych pomiarów. Będziemy mogli wówczas orzec, czy mamy do czynienia z tym samym rodzajem gumy, który uwzględniono w tablicach.

Wolniejszy wyprzedza

W wielu dyscyplinach sportu chodzi o to, by jak najszybciej dotrzeć do mety. By wygrać, trzeba dotrzeć do mety przed rywalami, co niekoniecznie oznacza ustanowienie nowego rekordu świata. Wydaje się więc oczywiste, że ten, kto ma większą prędkość, pierwszy ukończy bieg i wygra. Ale nie zawsze tak jest.

Właściwie każda bieżnia, tor łyżwiarski, żużlowy lub Formuły 1, a nawet ulica, na której rozgrywa się wyścig, ma zakręty. No właśnie, jak to jest z wyprzedzaniem na zakręcie?

Wyobraźmy sobie, że mamy dwóch zawodników ścigających się na torze z zakrętami. Każdy z nich ma takie same umiejętności techniczne, tzn. może pokonywać łuki przy takiej samej wartości przyspieszenia odśrodkowego (sytuacja taka dobrze odpowiada wyścigom F1). Oba wchodzą w zakręt w tym samym momencie. Jeden z nich jest bliżej środka łuku niż Drugi ($R_J < R_D$). Przyspieszenie odśrodkowe ma tę samą wartość, więc:

$$\omega_J^2 R_J = \omega_D^2 R_D,$$

gdzie ω_J to prędkość kątowa Jednego, a ω_D – Drugiego. Czytelnik Sprawny zaraz przekształci wzór i otrzyma:

$$\frac{\omega_J}{\omega_D} = \sqrt{\frac{R_D}{R_J}}.$$

Sprawa jest jasna – Jeden ma większą prędkość kątową, więc wyprzedza Drugiego. A jakie są ich prędkości liniowe? Czytelnik Sprawny musi sobie przypomnieć wzór wiążący prędkość liniową (V) z kątową i w mig dojdzie do wniosku, że:

$$\frac{V_J}{V_D} = \sqrt{\frac{R_J}{R_D}}.$$

Coś się odwrócił ten ułamek. Widzimy teraz, że wolniejszy wyprzedza!

Radosław POLESKI



Krople deszczu

Złota polska jesień się kończy. Nieuchronnie wchodzimy w okres slot jesiennych. Zamiast wpadać w minorowy nastrój, można się zastanowić, na ile rozumiemy zjawisko deszczu. Konkretnie chodzi o rozmiar kropel. Czy jest jakiś związek między ich wielkością a intensywnością deszczu?

Doświadczenie podpowiada nam, że ulewy rozpoczynają się od stosunkowo dużych kropel, których nie obserwujemy dla „kapuśniaczku”.

Te wrażenia można skonfrontować z systematycznymi badaniami, które po raz pierwszy zostały przeprowadzone 105 lat temu przez W. Bentleya oraz, niezależnie, przez P. von Lenarda. W 1948 roku natomiast J.S. Marshall i W.M. Palmer wykazali, że rozkład wielkości kropel jest wykładniczy

$$n(d) \sim e^{-d/\langle d \rangle},$$

gdzie d jest rozmiarem kropli, a średni rozmiar

$$\langle d \rangle \sim \mathcal{R}^{-0,21}$$

jest proporcjonalny do pewnej potęgi intensywności \mathcal{R} , która jest miarą tempa przyrostu słupa wody opadu (typowe jednostki to mm/h).

Żeby pogodzić tę obserwację z naszym doświadczeniem, wystarczy pamiętać o dwóch rzeczach. Po pierwsze, w celu oszacowania ilości wody przypadającej na dany rozmiar kropli należy rozkład wielkości pomnożyć przez objętość kropli ($V = \pi d^3/6$) oraz prędkość jej opadania (wynikającą z warunków równowagi sił oporu i grawitacji: $v \sim \sqrt{gd}$), co łącznie daje czynnik $d^{3+1/2}$. Powoduje to, że pojawia się wyraźne maksimum wielkości kropli mających wkład w sumaryczny opad. Maksimum to przypada dla tym większego rozmiaru kropli, im większa jest intensywność deszczu. Po drugie, samotne duże krople obserwujemy przede wszystkim na początku ulewy, czyli wtedy gdy spadają one na ziemię pojedynczo. Wtedy małe krople wyparowują w locie (intensywność parowania zmniejsza się wraz z gęstością kropel).

Okazuje się, że empirycznie stwierdzoną proporcjonalność między średnim rozmiarem kropli $\langle d \rangle$ a intensywnością \mathcal{R} można z powyższego rozumowania odtworzyć [1]. Wystarczy wprowadzić zmienną $x = d/\langle d \rangle$ i scałkować, żeby uzyskać związek

$$\mathcal{R} = \langle d \rangle^{9/2} \int f(x) dx \sim \langle d \rangle^{9/2},$$

a następnie zauważyć, że $9/2 = 0,21$.

Jest wiele naturalnych obiektów, których rozkład rozmiarów jest opisywany funkcją wykładniczą. Za każdą taką dystrybucję odpowiada jakiś mechanizm tworzenia się takich obiektów. Od dość dawna wiadomo, że w przypadku kropli deszczu takim zjawiskiem może być rozpad większych kropli na mniejsze. Uważano, że rozpad ten jest związany ze zderzeniami kropel, co jednak nie pozwalało na opracowanie zgadzającego się z doświadczeniem modelu, ze względu na zbyt niskie prawdopodobieństwo zderzeń.

Tym czasem, w pracy [1], wykazano, że wystarczy uwzględnienie rozpadu izolowanych kropel w czasie swobodnego ich opadania. Krople rosną w chmurach

do rozmiaru, dla którego ruchy konwekcyjne nie wystarczają do ich utrzymania, a parowanie w czasie spadania nie jest wystarczające do ich zniknięcia w locie.

W omawianej pracy za pomocą ultraszybkiej kamery wykonano zdjęcia określonej wielkości kropel umieszczanych w strumieniu powietrza. W artykule pokazane są serie zdjęć poklatkowych, a w materiałach dodatkowych zamieszczono również film. Podstawowy sposób rozpadu kropli jest następujący. Kropla najpierw spłaszcza się, zachowując grubszy brzeg, następnie z wnętrza takiego krążka tworzy się jakby czasza spadochronu i w końcu następuje dezintegracja na drobne kropelki różnych rozmiarów. Zliczanie tych kropelek pozwala na odtworzenie znanych prawidłowości, które zgadzają się z przedstawionymi obliczeniami dynamiki kropli w oparciu o zasady podstawowe.

„Po zrozumieniu, co dzieje się z pojedynczą, izolowaną kroplą, bez uwzględniania jakiegokolwiek oddziaływania z innymi kroplami, można zdać sobie sprawę, że produkty jej rozpadu wystarczają do ilościowego opisanego statystycznego składu deszczu. Tak więc oddziaływania nie są potrzebne w tym opisie. To jest podstawowa obserwacja” – powiedział Villermaux dziennikarzowi *Scientific American* [2].

Artykuł zawiera również obliczenia skal czasowych opisanych mechanizmów. Według autorów kolejnym wyzwaniem byłoby przeprowadzenie odpowiednich obserwacji rzeczywistego deszczu. Jesienią nie powinno zabraknąć po temu okazji.

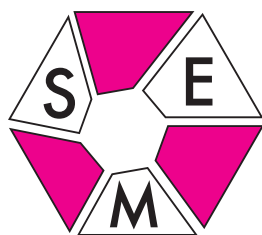
Piotr ZALEWSKI

[1] Emmanuel Villermaux i Benjamin Bossa, *Single-drop fragmentation determines size distribution of raindrops*, *Nature Physics*, 20/07/2009, DOI:10.1038/NPHYS1340

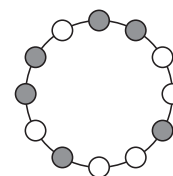
[2] John Matson, *Scientific American* 20/07/2009 <http://www.scientificamerican.com/article.cfm?id=raindrop-size-physics>

Stowarzyszenie na rzecz Edukacji Matematycznej

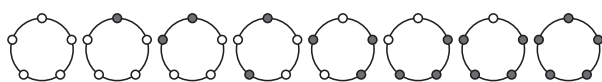
www.sem.edu.pl



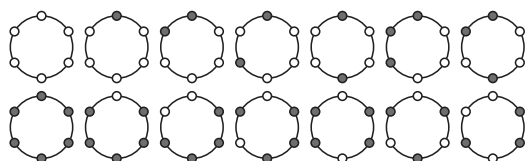
Odbicie rysunku okręgu w lustrze nie różni się niczym od tegoż rysunku. Jeśli jednak umieścimy na tym okręgu dwanaście kropek rozmieszczonych jak godziny na zegarze i niektóre z nich będą puste, a niektóre pełne, to będzie można zrobić to w ten sposób, by odbicie lustrzane tego rysunku dawało się od niego odróżnić – tak jest np. na rysunku obok (prawda?).



Czy takiej sztuczki nie da się zrobić z mniejszą liczbą punktów? Poniżej mamy wszystkie (na pewno?) 8 sposobów równomiernego umieszczenia pięciu białych i czarnych kropek na okręgu – jak widać, żaden z nich nie różni się od swego odbicia lustrzanego (a, mówiąc dokładniej, różni się najwyżej o obrót).

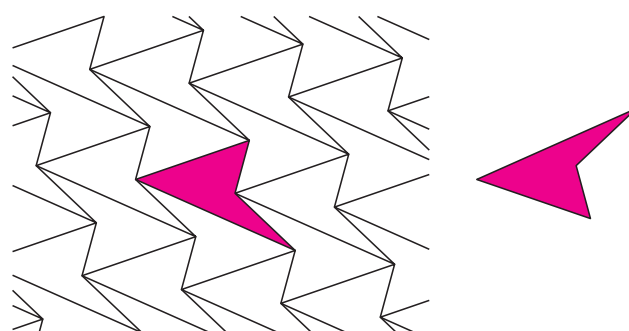


Natomiast wśród 14 sposobów równomiernego rozmieszczenia na okręgu sześciu białych i czarnych kropek będzie już jedna para (która?) zamieniająca się miejscami na lustrzanym odbiciu.

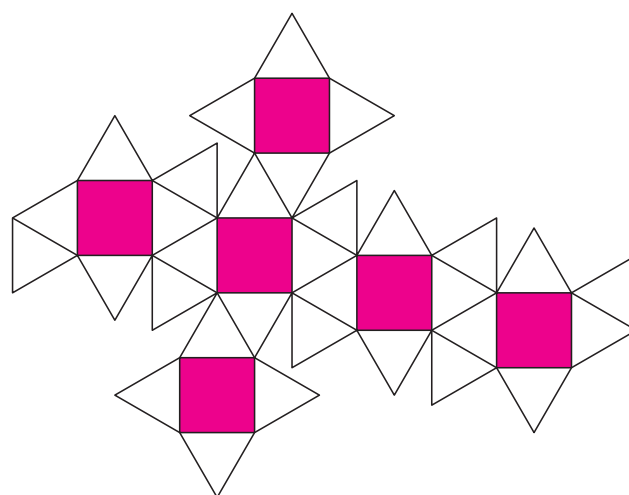


Trudność udzielenia odpowiedzi na pytanie, ile jest takich zamieniających się par np. dla dwunastu kropek, pokazuje nam, jak nieoczywiste jest – zdawałoby się naturalne – pojęcie **orientacji**, występujące właściwie we wszystkich gałęziach nauki (i często przedstawiane także za pomocą słów: **skrętność**, **chiralność**, **enancjomorficzność**). W technice mamy prawo- i lewoskrętne śruby, w zoologii także muszle, w genetyce (wszystkie?) helisy DNA mają tę samą skrętność, w chemii przyznano w 2001 roku Nagrodę Nobla za podanie sposobu syntezowania związków organicznych o określonej chiralności, w fizyce orientacja występuje w regule Oersteda czy sile Lorentza itd. itp.

Ale pozostajmy przy matematyce. Spośród wszystkich możliwych pięciu rytmów ornamentu liniowego (czy faktycznie jest ich pięć?), jakie można uzyskać z czcionki **R**, trzy niosą na sobie orientację (czyli różnią się od swego odbicia lustrzanego), a dwa nie – proszę sprawdzić.



Jeszcze ciekawiej jest w przestrzeni. Poniżej narysowana jest siatka bardzo regularnego wielościanu (w każdym wierzchołku zbiega się tyle samo takich samych wielokątów), która ma tę własność, że jeśli ją skleimy kolorowymi ścianami na zewnątrz, to da nam wielościan lustrzany, ale nie identyczny z tym, który otrzymamy, sklejając tę siatkę kolorem do wewnątrz. Ile jest takich, niosących orientację, wielościanów?



A przecież są i takie obiekty, które nie dość, że nie niosą na sobie orientacji, to nawet żadnej nieść nie mogą – np. wstęga Möbiusa.

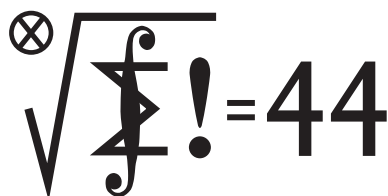
Orientacja to kolejne pole do ciekawej zabawy.

Klub 44

Liga zadaniowa Wydziału Matematyki, Informatyki i Mechaniki, Wydziału Fizyki Uniwersytetu Warszawskiego i Redakcji *Delta*

Skrót regulaminu

Każdy może nadsyłać rozwiązania zadań z numeru n w terminie do końca miesiąca $n + 2$. Szkice rozwiązań zamieszczamy w numerze $n + 4$. Można nadsyłać rozwiązania czterech, trzech, dwóch lub jednego zadania (każde na oddzielnej kartce), można to robić co miesiąc lub z dowolnymi przerwami. Rozwiązania zadań z matematyki i z fizyki należy przesyłać w oddzielnych kopertach, umieszczając na kopercie dopisek: **Klub 44 M** lub **Klub 44 F**. Oceniamy zadania w skali od 0 do 1 z dokładnością do 0,1. Ocenę mnożymy przez współczynnik trudności danego zadania: $WT = 4 - 3S/N$, gdzie S oznacza sumę ocen za rozwiązania tego zadania, a N – liczbę osób, które nadesłały rozwiązanie choćby jednego zadania z danego numeru w danej konkurencji (**M** lub **F**) – i tyle punktów otrzymuje nadsyłający. Po zgromadzeniu **44** punktów, w dowolnym czasie i w którejkolwiek z dwóch konkurencji (**M** lub **F**), zostaje on członkiem **Klubu 44**, a nadwyżka punktów jest zaliczana do ponownego udziału. Trzykrotne członkostwo – to tytuł **Weterana**. Szczegółowy regulamin został wydrukowany w numerze 2/2002 oraz znajduje się na stronie <http://www.mimuw.edu.pl/delta/regulamin.html>.



Termin nadsyłania rozwiązań: 31 XII 2009

Czołówka ligi zadaniowej **Klub 44M** po uwzględnieniu ocen rozwiązań zadań 575 ($WT = 2,18$) i 576 ($WT = 1,55$) z numeru 2/2009

Krzysztof Dorobisz	Kraków	47,00
Andrzej Idzik	Bolesławiec	44,70
Tomasz Warszawski	Kraków	42,26
Zbigniew Galias	Kraków	42,05
Paweł Najman	Jaworzno	38,98
Jerzy Cisło	Wrocław	35,48
Janusz Olszewski	Warszawa	34,01

Zamykając trzecią rundę, Krzysztof Dorobisz powiększył liczbę Weteranów Klubu 44M do 33 ($= \frac{3}{4} \cdot 44$).

Miejsce w Klubie matematycznym znalazł dla siebie także Andrzej Idzik, równoległe do sukcesów odnoszonych w Lidze fizycznej.

583. Oznaczmy badaną sumę przez s . Tak więc

$$\sum_{k \equiv 0} a_k = s \quad \text{oraz} \quad \sum_{k \not\equiv 0} a_k = P(1) - s;$$

wszystkie kongruencje w tym rozwiązaniu są brane (mod 3).

Niech α i β będą nierzeczywistymi pierwiastkami trzeciego stopnia z jedności:

$$\alpha = \frac{-1 + \sqrt{3}i}{2}, \quad \beta = \frac{-1 - \sqrt{3}i}{2}.$$

Ponieważ $\alpha^3 = \beta^3 = \alpha\beta = 1$, więc

$$\alpha^k = \begin{cases} 1 & \text{dla } k \equiv 0, \\ \alpha & \text{dla } k \equiv 1, \\ \beta & \text{dla } k \equiv 2, \end{cases} \quad \beta^k = \begin{cases} 1 & \text{dla } k \equiv 0, \\ \beta & \text{dla } k \equiv 1, \\ \alpha & \text{dla } k \equiv 2. \end{cases}$$

A skoro $\alpha + \beta = -1$, to

$$\alpha^k + \beta^k = \begin{cases} 2 & \text{dla } k \equiv 0, \\ -1 & \text{dla } k \not\equiv 0. \end{cases}$$

Stąd

$$\begin{aligned} P(\alpha) + P(\beta) &= \sum_{k \equiv 0} a_k(\alpha^k + \beta^k) + \sum_{k \not\equiv 0} a_k(\alpha^k + \beta^k) = \\ &= 2s - (P(1) - s) = 3s - 4^{44}. \end{aligned}$$

Zadania z matematyki nr 587, 588

Redaguje Marcin E. KUCZMA

587. Dana jest liczba dodatnia $a \leq 1/2$. Określamy ciąg (x_n) wzorami:

$$x_1 = 1, \quad x_{n+1} = x_n - ax_n^2 \quad \text{dla } n \geq 1.$$

Dowieść, że ciąg (nx_n) jest zbieżny, i obliczyć jego granicę.

588. Wykazać, że dla każdej liczby naturalnej $n \geq 1$ równanie

$$x^2 + xy + y^2 = 13^n$$

ma rozwiązanie w liczbach całkowitych x, y .

Zadanie 588 zaproponował pan Witold Bednarek z Łodzi.

Rozwiązania zadań z numeru 6/2009

Przypominamy treść zadań:

583. Rozważamy wielomian

$$P(x) = (1 + x^{43} + x^{44} + x^{45})^{44} = a_0 + a_1x + \dots + a_{1980}x^{1980}.$$

Obliczyć sumę $a_0 + a_3 + a_6 + \dots + a_{1980}$ tych współczynników, których numery dzielą się przez 3.

584. Dla liczb dodatnich a, b, c przyjmijmy

$$\begin{aligned} U &= a^2b + b^2c + c^2a, & V &= ab^2 + bc^2 + ca^2, \\ A &= a^3 + abc, & B &= b^3 + abc, & C &= c^3 + abc. \end{aligned}$$

Udowodnić, że $\sqrt{UV} \geq abc + \sqrt[3]{ABC}$.

Z drugiej strony,

$$P(\alpha) = (1 + \alpha + \beta + 1)^{44} = 1,$$

i tak samo

$$P(\beta) = 1.$$

Z uzyskanych związków wyznaczamy $s = (4^{44} + 2)/3$.

584. Wobec jednorodności (sześcienniej) wyrażeń A, B, C, U, V można przyjąć, że $abc = 1$. Nietrudno sprawdzić, że wówczas

$$UV = W + 3 \quad \text{oraz} \quad ABC = W + 2,$$

gdzie

$$W = a^3b^3 + b^3c^3 + c^3a^3 + a^3 + b^3 + c^3.$$

Skoro $abc = 1$, to $W \geq 3 + 3 = 6$ (nierówności średnich). Przepisujemy więc nierówność z tezy zadania jako

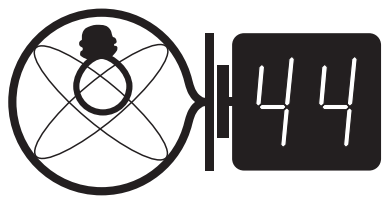
$$\sqrt{W + 3} \geq 1 + \sqrt[3]{W + 2}.$$

Przyjmijmy oznaczenie $X = \sqrt[3]{W + 2}$; zatem $X \geq 2$. Dowiedziona nierówność przybiera postać

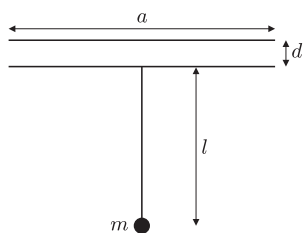
$$\sqrt{X^3 + 1} \geq X + 1;$$

równoważnie: $X^2 \geq X + 2$. Dla $X \geq 2$ jest to prawda.

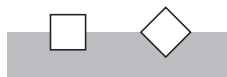
Klub 44



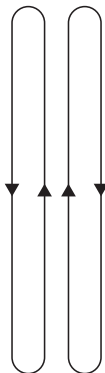
Termin nadsyłania rozwiązań: 31 XII 2009



Rys. 1



Rys. 2



Rys. 3

Zadania z fizyki nr 484, 485

Redaguje Jerzy B. BROJAN

484. Okładki kondensatora są prostokątami o wymiarach a i b , a odległość między ich środkami jest równa d , przy czym $d \ll a$, $d \ll b$. Górna okładka jest pozioma i nieruchoma, a dolna może się obracać wokół osi równoległej do boku b i przechodzącej przez jej środek. Dolna okładka jest sztywno połączona z prętem o długości l , na końcu którego znajduje się ciężarek o masie m (rys. 1). Naładowano kondensator do napięcia U . Jaki warunek muszą spełniać wymienione parametry, aby pionowe położenie pręta (kiedy okładki są równoległe) było położeniem równowagi trwałej?

485. Mamy trzy jednakowe ciała, których ciepło właściwe jest stałe (nie zależy od temperatury), a temperatury wynoszą 100 K, 200 K i 300 K. Do jakiej maksymalnej temperatury można ogrzać jedno z tych ciał, jeśli energia przepływa tylko między nimi, przy czym możemy wykorzystywać maszyny ciepłe?

Rozwiązania zadań z numeru 6/2009

Przypominamy treść zadań:

480. Powierzchnia długiego walca o promieniu r jest przewodząca. Po tej powierzchni wzdłuż osi walca płynie prąd o natężeniu I , o gęstości stałej wzdłuż obwodu walca. Jakie ciśnienie p powstaje wewnątrz walca wskutek wzajemnego przyciągania się elementów powierzchni? Względna przenikalność magnetyczna ośrodka jest równa 1.

481. Długi jednorodny pręt o przekroju kwadratowym, wykonany z materiału o gęstości $0,5 \text{ g/cm}^3$, leży na powierzchni wody. Która z przedstawionych na rysunku 2 pozycji jest położeniem równowagi trwałej, czy też obie?

480. Rozważmy przemianę energii przy niewielkiej zmianie promienia walca. Aby uniknąć uwzględniania energii dostarczonej przez źródło zasilania, przyjmijmy, że obwód jest zamknięty zewnętrznym walcem o promieniu R (zob. rys. 3) i bezporowoty – wtedy strumień pola magnetycznego Φ pozostaje stały, a praca objętościowa jest równa zmianie energii pola

$$pdV = pl \cdot 2\pi r dr = dE_m,$$

gdzie l jest długością walców. Pole magnetyczne występuje tylko w obszarze między walcami, jego indukcja jest dana wzorem

$$B = \frac{\mu_0}{2\pi} \frac{I}{r'}, \quad r < r' < R,$$

a strumień obliczamy jako całkę

$$\Phi = l \int_r^R B dr' = \frac{\mu_0}{2\pi} l I \ln(R/r).$$

Z warunku $\Phi = \text{const}$ wyprowadzamy związek między niewielkimi przyrostami dr i dI :

$$\ln(R/r)dI = (I/r)dr.$$

Gęstość energii pola magnetycznego w próżni jest dana wyrażeniem $B^2/2\mu_0$, zatem całkowitą energię pola w obszarze między walcami obliczamy jako

$$E_m = \frac{1}{2\mu_0} \int B^2 dV = \frac{2\pi l}{2\mu_0} \int_r^R B^2 r' dr' = \frac{\mu_0}{4\pi} l I^2 \ln(R/r).$$

Uwzględniając związek między przyrostami dr i dI , obliczamy

$$dE_m = \frac{\mu_0 l I^2}{4\pi r} dr, \quad \text{a stąd} \quad p = \frac{\mu_0}{8\pi^2} \frac{I^2}{r^2}.$$

Zauważmy, że pomijając warunek $\Phi = \text{const}$ i wynikającą z niego zmianę natężenia prądu, otrzymalibyśmy wynik pozornie prawidłowy, ale w istocie błędny: wzrost promienia r oznaczałby bowiem zmniejszenie obszaru pola magnetycznego i zmniejszenie jego energii. To by oznaczało, że elementy powierzchni walca nie powinny się przyciągać, lecz odpychać (niezgodnie z rzeczywistością).

481. W obu pozycjach środek masy prętów leży na tej samej wysokości (na powierzchni wody), więc energia grawitacyjna prętów jest jednakowa. Dla lewej pozycji środek masy części zanurzonej leży jednak niżej, czyli środek masy wody – wyżej. Energia grawitacyjna wody jest niższa dla prawej pozycji, zatem to ona przedstawia równowagę trwałą.

Ściśle rzecz biorąc, samo porównanie dwóch energii potencjalnych nie rozstrzyga jednoznacznie kwestii rodzaju położenia równowagi (energia mogłaby być dowolnie skomplikowaną funkcją kąta przechyłu, z większą liczbą ekstremów). Bardziej precyzyjną metodę analizy równowagi ciał pływających przedstawiono w rozwiązaniu zadania 223, *Delta* 1/1997.

Czołówka ligi zadaniowej **Klub 44F**
po uwzględnieniu ocen rozwiązań zadań
474 ($WT = 2,56$) i 475 ($WT = 2,02$)
z numeru 3/2009

Tomasz Wietecha	Tarnów	42,24
Andrzej Idzik	Bolesławiec	37,42
Krzysztof Magiera	Łosiów	30,37
Andrzej Nowogrodzki	Chocianów	28,18
Radosław Poleski	Kołobrzeg	23,47
Michał Koźlik	Gliwice	18,13



Rozwiązanie zadania F 750.

Ze względu na wzajemne położenie ekranu, źródła światła oraz zwierciadła, „zajaczek” będzie miał kształt trójkąta, podobnego do trójkąta zwierciadła. Wszystkie liniowe rozmiary „zajaczka” będą przy tym dwa razy większe od odpowiednich rozmiarów zwierciadła. Zatem pole powierzchni S będzie cztery razy większe:

$$S = 4s,$$

gdzie pole powierzchni s zwierciadła wynosi

$$s = \frac{a^2\sqrt{3}}{4}.$$

Ostatecznie

$$S = a^2\sqrt{3} \approx 1,7 \cdot 10^{-2} \text{ m}^2.$$



Rozwiązanie zadania M 1257.

Niech $[F]$ oznacza pole figury \mathcal{F} . Zauważmy, że $\sphericalangle NAS = 180^\circ - \sphericalangle BAC$. A zatem

$$\begin{aligned} [NAS] &= \frac{1}{2} \cdot AN \cdot AS \cdot \sin(\sphericalangle NAS) = \\ &= \frac{1}{2} \cdot AC \cdot AB \cdot \sin(\sphericalangle BAC) = [ABC]. \end{aligned}$$

Analogicznie dowodzimy, że pola trójkątów BRQ oraz MPC są równe polu trójkąta ABC , co kończy dowód.

Patrz w niebo

Gdybyśmy mogli oglądać niebo w zakresie rentgenowskim, to na wielkim obszarze od Bliźniąt do Jednorożca byłoby widać pierścieni o średnicy 25° , zwany *Monogem Ring* – od nazw gwiazdozbiorów Monoceros i Gemini. W październiku tę część nieba widać najlepiej nad ranem, ale nie musimy wstawać po nocy, bo i tak nie mamy rentgenowskich oczu. Takie obiekty kojarzą się z mgławicami, będącymi pozostałościami po supernowych. Skojarzenie wszakże nie wystarcza – należy uzyskać jakieś za tym argumenty, a pomogły w tym sztuczne satelity. Bez nich samo stwierdzenie obecności takiego pierścienia byłoby niemożliwe. Ziemska atmosfera nie przeszkadza wprawdzie w stosowaniu promieniowania rentgenowskiego w medycynie, ale kilkaset kilometrów atmosfery całkiem skutecznie przesłania naziemnemu obserwatorowi rentgenowskie niebo. Zresztą, całe szczęście. W każdym razie odległość pierścienia Monogem oszacowano na 300 pc, a jego wiek na 100 000 lat. Supernowa w takiej odległości w maksimum blasku miałaby jasność porównywalną z jasnością Księżyca.

Do kompletu zaobserwowano też blisko centrum mgławicy pulsara, a niezależne oceny jego odległości zgadzają się z odległością mgławicy. Jego temperaturę oceniono na 840 000 K, a średnicę na nie więcej niż 40 km. Badacze dość zgodnie podejrzewają, że cały ten złożony obiekt (pulsar plus mgławica) jest najważniejszym źródłem promieni kosmicznych docierających do Ziemi. Hipotezę, według której wysokoenergetyczne cząstki padające na Ziemię pochodzą w większości z jednego źródła, wysunięto już w latach 90. ubiegłego wieku, a było to przed odkryciem i zlokalizowaniem wspomnianego pulsara. Teraz źródło to w zasadzie jest znane, ale stuprocentowej pewności nie ma. Nie zapominajmy bowiem, że kierunek ruchu cząstek przy Ziemi nie musi celować w ich źródło, a to wskutek oddziaływania ich podczas kosmicznej podróży z wszechobecnym galaktycznym polem magnetycznym.

Tomasz KWAST

Październik

Trochę to dziwne, że Pegaza starożytni astronomowie widzieli na niebie do góry nogami, w każdym razie tak go widać z północnej półkuli Ziemi. Co prawda w październikowe wieczory przechodzi dość blisko zenitu, ale po jego południowej stronie i wtedy do horyzontu skierowany jest głową. Skoro jednak w Kosmosie pojęcia góry i dołu są nieokreślone, jest to bez znaczenia. W Pegazie, choć jest to duży gwiazdozbiór, nie ma, niestety, interesujących obiektów dostępnych dla amatora. Najjaśniejsza w nim gromada kulista ma jasność 6,3 mag, a galaktyka 11,2 mag. Okazały prostokąt utworzony przez najjaśniejsze gwiazdy Pegaza służy więc miłośnikom astronomii jako układ odniesienia do odnajdywania innych pobliskich gwiazdozbiorów, np. na północny wschód sąsiedniej Andromedy, gdzie nieuzbrojonym okiem widać najjaśniejszą galaktykę na niebie, M31. Najjaśniejsza gwiazda Pegaza (2,54 mag) oznaczona jest aż piątą literą greckiego alfabetu, epsilon.

Merkury znajdzie się 6 X najdalej od Słońca i można go szukać na wschodnim niebie, zanim wzejdzie Słońce. Wenus i Saturn są w Pannie, gdzie jest też Słońce, zatem tych planet nie widać. Mars jest w Raku i widać go w drugiej połowie nocy, a Jowisz w Koziorożcu i widać go w pierwszej połowie nocy. Saturn i Merkury zejdą się 8 X na $0^\circ 3'$, ale i tak nastąpi to zbyt blisko Słońca. Wreszcie Saturn i Wenus zejdą się 13 X na $0^\circ 6'$, czego też z powodu bliskości Słońca nie da się zobaczyć. Pełnia Księżyca wypada 4 X, a nów 18 X. Księżyc zakryje Marsa 12 X, co będą mogli zobaczyć podróżujący akurat po Oceanie Indyjskim. Zakryje też Antaresa 21 X, co w zasadzie będzie widoczne w Europie, ale będzie wtedy bardzo wczesny wieczór. Będzie to zarazem ostatnie zakrycie Antaresa w serii comiesięcznych zakryć trwającej od wielu miesięcy. W październiku można spodziewać się dwóch słabych rojów meteorów: Giacobinidów około 9 X i Orionidów 20 X.

T. K.



W każdej tego typu skończonej grze bez remisów, bez elementów losowych (typu rzuty kostką), w której obaj gracze mają pełną wiedzę (żaden, na przykład, nie ukrywa swoich kart), dla któregoś z nich istnieje strategia wygrywająca.

Gry

Joanna JASZUŃSKA

W wielu grach dla któregoś z graczy istnieje *strategia wygrywająca*, czyli taka „recepta” na grę, która pozwala zawsze zwyciężyć, niezależnie od ruchów przeciwnika. Jednak strategię taką, nawet jeśli istnieje, nie zawsze łatwo wskazać. Na szczęście często można. Czasem wystarczą do tego proste pomysły typu symetria, czasem zaś potrzebne są metody bardziej wyrafinowane. W niektórych grach nawet bez żadnej strategii wynik jest z góry przesądzony. Ilustrują to poniższe przykłady.

Gracza rozpoczynającego oznaczamy G_I , drugiego G_{II} . Wykonują oni ruchy na przemian, według podanych reguł. Jeśli nie jest powiedziane inaczej, przegrywa gracz, który nie może wykonać ruchu. Należy rozstrzygnąć, czy któryś z graczy ma strategię wygrywającą (lub przynajmniej nieprzegrywającą, gdy gra dopuszcza remisy), i wskazać ją.

1. Na okrągłym stoliku gracze kładą złotówki, przy czym nie mogą one wystawać poza stolik ani nachodzić na siebie oraz nie wolno przesuwać leżących już monet.

R. Wygrywa G_I : umieszcza pierwszą złotówkę na środku stolika, po czym na każdy ruch przeciwnika odpowiada, kładąc monetę symetrycznie względem środka. Jeśli G_{II} znalazł miejsce na monetę, to G_I też znajdzie – miejsce symetryczne jest wolne. \square

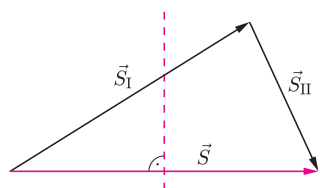
2. Gracze rysują takie przekątne 12-kąta foremnego, które się nie przecinają.

R. Wygrywa G_I . Może zacząć od narysowania jednej z głównych przekątnych i na każdy ruch przeciwnika odpowiadać ruchem symetrycznym względem niej. \square

Co więcej, G_I wygrywa nawet bez żadnej strategii, gdyż gra zawsze trwa 9 ruchów. Zauważ, że n -kąta foremny ma $n - 3$ nieprzecinające się przekątne.

3. Na płaszczyźnie danych jest 20 wektorów. Gracze wybierają po jednym z nich. Wygrywa ten, kto na końcu ma dłuższą sumę wybranych wektorów.

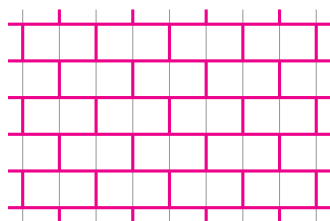
R. Oznaczmy przez \vec{S}_I i \vec{S}_{II} sumy wektorów wybranych przez graczy, a przez \vec{S} sumę danych wektorów: $\vec{S} = \vec{S}_I + \vec{S}_{II}$. Jeśli $\vec{S} = 0$, to gra kończy się remisem, bo $\vec{S}_I = -\vec{S}_{II}$. Jeśli $\vec{S} \neq 0$, to wektory \vec{S} , \vec{S}_I i \vec{S}_{II} tworzą trójkąt (rys. 1) – być może zdegenerowany. Stąd $|\vec{S}_I| > |\vec{S}_{II}|$ wtedy i tylko wtedy, gdy koniec wektora \vec{S}_I znajduje się po tej samej stronie symetralnej wektora \vec{S} , co koniec \vec{S} . Strategią dla G_I jest zatem maksymalizowanie składowej \vec{S}_I w kierunku \vec{S} , czyli spośród dostępnych wektorów branie zawsze takiego o największej składowej w kierunku \vec{S} . Wygrywa lub remisuje, bo G_{II} bierze zawsze wektor o co najwyżej równie dużej składowej w tym kierunku i łącznie ma tyle samo wektorów. Gdy gra jest remisowa, żadna inna strategia nie zagwarantowałaby G_I wygranej. \square



Rys. 1

4. Gracze stawiają pionki, po jednym, na polach nieskończonej szachownicy. Gracz wygrywa, jeżeli utworzy ze swoich pionków kwadrat 2×2 .

R. Pokolorujemy szachownicę w mur (rys. 2). Każdy kwadrat 2×2 zawiera w całości pewną cegłę 2×1 . Jeśli G_{II} zawsze wstawia swój pionek, złośliwie, do tej samej cegły, w której właśnie zagrał G_I , to G_I nie może wygrać. Jest to więc strategia nieprzegrywająca dla G_{II} . Pozostawiam Czytelnikom podobne sprawdzenie, że G_{II} nie ma strategii wygrywającej. \square



Rys. 2

Zadania domowe:

5. Planszą do gry są wierzchołki trójkąta oraz 7 punktów z jego wnętrza, żadne trzy punkty planszy nie są współliniowe. Gracze rysują nieprzecinające się odcinki łączące dwa z danych 10 punktów.

Wskazówka. Po grze plansza staje się grafem planarnym o trójkątnych ścianach. Wykorzystaj fakt, że $2k = 3s$, oraz wzór Eulera $w - k + s = 2$, gdzie $w = 10$ to liczba punktów planszy, k – liczba narysowanych odcinków, zaś s – liczba ścian (wliczając też ścianę zewnętrzną).

6. Dana jest tablica 2008×2008 . Dwaj gracze na przemian wykonują ruchy, z których każdy polega na wybraniu białego albo czarnego pionka i postawieniu go na wybranym wolnym polu. Wygrywa ten, którego ruch doprowadził do powstania ciągu 5 kolejnych pionków tego samego koloru w linii pionowej, poziomej lub ukośnej. Zbadaj, czy dla gracza rozpoczynającego grę istnieje strategia zapewniająca mu zwycięstwo.

Zadanie pochodzi z tegorocznej, LX Olimpiady Matematycznej. Rozwiązanie na stronie www.om.edu.pl