Strategia w grze w kamyki

Dagna CZUBLA, Marcin WIERZBIŃSKI***

* Uczennica, XIV LO im. Stanisława Staszica w Warszawie

** Doktorant, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

Niektóre problemy informatyczne można rozwiązać prostymi algorytmami, których poprawność (tzn. udzielanie odpowiedzi zgodnie z oczekiwaniami) jest oczywista. Istnieją też problemy, do których rozwiązania potrzebne są bardzo skomplikowane algorytmy, a znane dowody ich poprawności są wyjątkowo złożone i zagmatwane. W tym artykule będziemy się zajmować problemem, który może być rozwiązany przez bardzo prosty algorytm, ale udowodnienie jego poprawności wymaga już pewnego wysiłku.

Na tegorocznej próbnej maturze z informatyki pojawiło się ciekawe zadanie o nazwie *Gra w kamyki.* Należało podać efektywny algorytm obliczający, który gracz ma strategię wygrywającą. Okazuje się, że zadanie to można rozwiązać bardzo prostym algorytmem, ale formalny dowód jego poprawności wymaga więcej wysiłku. W czasie egzaminu samo podanie algorytmu wystarczało do uzyskania maksymalnej liczby punktów, a dowód poprawności nie był wymagany. W tym artykule przedstawimy wzorcowy algorytm oraz uzasadnimy jego poprawność.

Gra w kamyki

Ada i Bajtek postanowili zagrać w następującą grę. Na stole przed sobą rozłożyli n kamyków. Zasady gry są proste. Gracze wykonują ruchy na przemian, rozpoczyna Ada. W swoim ruchu gracz może zabrać ze stołu 1, 3 lub 4 kamyki. Gracz, który weźmie ostatni kamyk, wygrywa.

Przykładowy przebieg rozgrywki dla $n = 5$ *kamyków:*

Ada bierze jeden kamyk ze stołu, następnie Bajtek bierze cztery kamyki i wygrywa grę. Ada może też zacząć od zabrania trzech kamyków, następnie Bajtek może jedynie zabrać jeden kamyk (ponieważ na stole zostały dwa), a na końcu Ada zabiera ostatni kamyk i wygrywa grę.

Ada zastanawia się, dla jakich n ma strategię wygrywającą. To znaczy chciałaby wiedzieć, czy niezależnie od ruchów Bajtka będzie w stanie z nim wygrać, jeśli będzie mądrze wybierać swoje posunięcia. Okazuje się, że ten problem rozwiązuje bardzo prosty algorytm podany poniżej.

Algorytm 1. Sprawdzenie strategii wygrywającej dla Ady

```
procedure CzyAdaMaStrategięWygrywającą(n)
    if n mod 7 = 0 or n mod 7 = 2 then
        return „NIE”
    else
        return „TAK”
```

Algorytm 1 sprawdza, czy reszta z dzielenia liczby n przez 7 jest równa 0 lub 2. Jeśli tak, to zwraca „NIE” (czyli Ada nie ma strategii wygrywającej). W przeciwnym przypadku zwraca „TAK” (czyli Ada ma strategię wygrywającą). Uzasadnimy teraz formalnie poprawność tego algorytmu.

Zastanówmy się, co dzieje się dla małych wartości n. Jeżeli n jest równe $1, 3$ lub 4, to Ada może wziąć wszystkie kamyki ze stołu i wygrać – ma strategię wygrywającą. Jeśli $n = 2$, Ada musi wziąć 1 kamyk, zostawiając na stole 1 kamyk, który Bajtek musi wziąć, tym samym wygrywając; Ada nie ma w tym przypadku strategii wygrywającej. Jeśli $n = 5$ lub $n = 6$, to Ada może wziąć odpowiednio 3 lub 4 kamyki, zostawiając Bajtkowi 2 kamyki, Bajtek więc musi przegrać – Ada ma strategię wygrywającą w obu przypadkach. Jeżeli $n = 7$, Ada po swoim ruchu zostawi Bajtkowi $3, 4$ lub 6 kamyków na stole, Bajtek ma więc strategię wygrywającą. Stąd wynika, że dla $n = 7$ Ada nie ma strategii wygrywającej.

Tabelka dla początkowych liczb kamyków z określeniem, czy Ada ma strategię pozwalającą jej wygrać z Bajtkiem.

n	Strategia wygrywająca dla Ady?
1	TAK
2	NIE
3	TAK
4	TAK
5	TAK
6	TAK
7	NIE
8	TAK

Rozwiązanie zadania F 1083.
Przed ociepleniem przez jednorodną, płaską warstwę ciała stałego (ściany) o powierzchni A, której powierzchnie zewnętrzne utrzymywane są w różnych (ale stałych w czasie) temperaturach $T_1 < T_2$, przepływa ciepło o mocy:

$$\frac{dQ}{dt} = P_1 = \frac{\kappa_1}{d_1}(T_2 - T_1)A.$$

Po ociepleniu ściana składa się z dwóch jednorodnych warstw: styropianu o grubości d_2 i cegły o grubości d_1. Na granicy pomiędzy styropianem i cegłą ustali się temepratura T_3, $T_1 < T_3 < T_2$. Przez każdą z warstw (styropianu i cegły) przepływa taka sama ilość ciepła, ale teraz o mocy $P_2 \neq P_1$. Z warunku równości mocy przepływających przez każdą z warstw ściany otrzymujemy:

$$P_2 = \frac{\kappa_1}{d_1}(T_2 - T_3)A = \frac{\kappa_2}{d_2}(T_3 - T_1)A,$$

który pozwala wyznaczyć wartość

$$T_3 = \frac{\kappa_1 T_2/d_1 + \kappa_2 T_1/d_2}{\kappa_1/d_1 + \kappa_2/d_2},$$

a następnie

$$P_2 = \frac{\kappa_1\kappa_2(T_2 - T_1)/(d_1 d_2)}{\kappa_1/d_1 + \kappa_2/d_2} = \frac{\kappa_2/d_2}{\kappa_1/d_1 + \kappa_2/d_2} P_1.$$

Dla podanych danych $P_2 \approx 0{,}45 P_1$, czyli strata ciepła jest ponad 2,2 raza mniejsza niż przed ociepleniem. Jeśli przyjąć, że ściana ma dosyć typowe wymiary: 2,7 m na 4 m, a różnica temperatur $T_2 - T_1 = 20$ K, to $P_1 \approx 864$ W, a $P_2 \approx 391$ W.

Rozwiązanie zadania M 1762.
Odpowiedź: tak.
Niech pola lewe dolne i prawe górne leżą na głównej przekątnej szachownicy i mają „współrzędne" $(1,1)$ i $(100,100)$. Korzystając z zasady indukcji, udowodnimy, że można dostać się do dowolnego wolnego pola na tej przekątnej.
Rzeczywiście, załóżmy, że możemy przejść do pola (n,n). Jeżeli pole $(n+1,n+1)$ jest wolne, to co najmniej jedno z pól $(n,n+1)$ i $(n+1,n)$ nie jest zajęte i można przez nie przejść do pola $(n+1,n+1)$.
Jeżeli natomiast pole $(n+1,n+1)$ jest zajęte, to dokładnie jeden z jego sąsiadów jest zajęty, więc jedna z dwóch ścieżek od (n,n) do $(n+2,n+2)$ jest możliwa do przejścia.
Pytanie: czy dla wymiaru planszy 100×101 teza również zachodzi?

Ogólnie, jeśli Ada ma strategię wygrywającą dla $n-1, n-3$ oraz $n-4$ kamyków na początku, to nie ma strategii wygrywającej dla n kamyków, gdyż każdy jej ruch doprowadzi do sytuacji, w której Bajtek ma strategię wygrywającą.

Formalny dowód poprawności algorytmu będzie wykorzystywał zasadę indukcji matematycznej, o której można przeczytać na przykład w Kąciku Początkującego Olimpijczyka w Δ^7_{19}. Mianowicie: wiemy już, że dla $n \leqslant 7$ Ada nie ma strategii wygrywającej, jeśli n daje resztę 0 lub 2 z dzielenia przez 7, zaś w przeciwnym przypadku ma taką strategię. Pokażemy teraz, że jeśli ta własność zachodzi dla wszystkich $n \leqslant 7k$ dla pewnej liczby naturalnej k, to zachodzi również dla wszystkich n ze zbioru $\{7k+1, 7k+2, \ldots, 7k+7\}$.

Dla $n = 7k+1, n = 7k+3$ oraz $7k+4$ Ada może wziąć ze stołu odpowiednio $1, 3$ lub 4 kamyki. W konsekwencji na stole zostanie $7k$ kamyków i, jak wynika z naszego założenia, oznacza to przegraną Bajtka.

Jeśli $n = 7k+2$, to po ruchu Ady na stole będzie $7k+1, 7(k-1)+6$ albo $7(k-1)+5$ kamyków. Wiemy jednak z naszego założenia, że dla każdej z tych liczb Ada ma strategię wygrywającą, czyli nie ma strategii wygrywającej dla $n = 7k+2$.

Teraz jeśli $n = 7k+5$ lub $n = 7k+6$, to Ada może wziąć odpowiednio 3 lub 4 kamyki, po czym na stole zostaną $7k+2$ kamyki, więc, jak pokazaliśmy przed chwilą, Bajtek będzie na straconej pozycji.

W końcu jeśli na początku mamy $n = 7k+7$, to po ruchu Ady na stole będzie $7k+3, 7k+4$ albo $7k+6$ kamyków. Udowodniliśmy już, że w każdej z tych sytuacji Ada ma strategię wygrywającą, więc dla wyjściowego n Ada nie ma strategii wygrywającej. Zasada indukcji matematycznej kończy dowód poprawności algorytmu.

Co dalej?

Czy to, że gracze mogli zabrać $1, 3$ lub 4 kamyki, miało kluczowe znaczenie dla istnienia algorytmu o prostej strukturze stwierdzającego, kto ma strategię wygrywającą? Jak zmieniłaby się odpowiedź, gdyby gracze mogli zabierać na przykład $1, 4, 5$ lub 10 kamyków? Aby odpowiedzieć na te pytania, rozważmy ogólniejszy przypadek – każdy z dwojga graczy w swoim ruchu może zabrać ze stołu $k_1, k_2, \ldots, k_{l-1}$ lub k_l kamyków (przy czym liczby te są parami różnymi dodatnimi liczbami całkowitymi ustawionymi w kolejności malejącej). Tak jak wcześniej, przegrywa gracz, który nie może wykonać ruchu.

Rozważmy ciąg $(a_i)_{i\in\mathbb{N}}$, zdefiniowany następująco: a_n jest równe 1, jeśli Ada ma strategię wygrywającą przy n kamykach znajdujących się początkowo na stole oraz 0 w przeciwnym przypadku. Zauważmy, że dowolnych k_1 kolejnych wyrazów ciągu (a_i) determinuje wszystkie przyszłe jego wyrazy. Istotnie, jeśli wiemy, czy Ada ma strategię wygrywającą dla $n-k_1, n-k_2, \ldots, n-k_l$, to możemy stwierdzić, czy ma również strategię wygrywającą dla n.

Skoro różnych możliwych ciągów binarnych o długości k_1 jest 2^{k_1}, to od pewnej pozycji $p \leqslant 2^{k_1}$ wyrazy ciągu a_n zaczną się cyklicznie powtarzać, z pewnym okresem $q \leqslant 2^{k_1}$.

Stąd wynika, że nasz problem da się rozwiązać algorytmem o bardzo prostej strukturze, który unika jakichkolwiek pętli, a wykonuje maksymalnie jedno dzielenie modulo i pewną liczbę porównań. Taki algorytm powinien działać następująco:

- jeśli $n \leqslant p$, to algorytm ma zakodowaną odpowiedź dla n;
- jeśli $n > p$, to algorytm sprawdza resztę z dzielenia n przez q i na tej podstawie stwierdza, czy Ada ma strategię wygrywającą.

Zauważmy, że nie pokazaliśmy w związku z tym, jak konkretnie napisać algorytm dla ustalonych wartości $k_1, k_2, \ldots, k_l$ – wykazaliśmy tylko, że algorytm o prostej strukturze istnieje.

Czytelnik znający pojęcie deterministycznego automatu skończonego i języka regularnego (można o nich przeczytać w artykułach Wojciecha Czerwińskiego w Δ^9_{18} i Marii Donten-Bury w Δ^{11}_{10}) może zastanowić się nad następującym problemem: czy dla ustalonych $k_1, k_2, \ldots, k_l$ i jednoliterowego alfabetu $\{a\}$ język $L = \{a^n :$ Ada ma strategię wygrywającą dla n kamyków$\}$ jest regularny? Czy to, że języki regularne są rozpoznawane przez deterministyczne automaty skończone, pozwala jakoś inaczej wywnioskować, że nasz problem można rozwiązać algorytmem o prostej strukturze?

Jak widać, nawet zadania maturalne potrafią prowadzić ku zagadnieniom o głębokim charakterze teoretycznym. To pokazuje, jak bogaty i zaskakujący może być świat nauki, który kryje się za pozornie prostymi problemami.