Praktyczne, bo niepraktyczne

czyli: od teorii liczb do kryptologii

Tomasz KAZANA

Teoria liczb to prawdopodobnie najstarsza dziedzina wiedzy matematycznej, badana intensywnie już w czasach starożytnych (a zapewne jeszcze wcześniej; możliwe, że nawet zanim powstała jakakolwiek cywilizacja operująca przekazem pisemnym).

Greków niebywale fascynowały liczby pierwsze (umieli udowodnić, że jest ich nieskończenie wiele!) i różne zagadnienia obliczeniowe z nimi związane: potrafili na przykład sprawnie wyznaczać zbiór wszystkich liczb pierwszych nie większych niż N, dla zadanego N (sito Eratostenesa) czy szybko odnajdować największy wspólny dzielnik (NWD) dwóch liczb naturalnych (algorytm Euklidesa).

W zasadzie przez wszystkie kolejne stulecia zagadnienia z tej dziedziny były zawsze w orbicie zainteresowań matematyków, choć niespecjalnie miało to przełożenie na praktyczne zastosowania. Oczywiście dla rasowego matematyka nigdy nie jest to przeszkodą, a wręcz – co szczególnie podkreślał Carl Gauss – może stanowić, trudną do zrozumienia dla profanów matematyki, dodatkową motywację do ich zgłębiania.

Historia rozwoju teorii liczb jest fascynująca i pełna nieoczekiwanych wyników, jednak ambicją tego tekstu nie jest nawet jej naszkicowanie. Raczej – idąc z duchem numeru *Delty*, który Czytelnik trzyma w dłoni – chcemy pokazać pewne pojęcia, koncepcje i hipotezy, które znali już Fermat, Gauss, Goldbach czy Euler, a które okazały się użyteczne dopiero po ich śmierci. Innymi słowy, *królowa matematyki* (że pozwolimy sobie przywołać po raz drugi słowa Gaussa na temat teorii liczb) nie jest aż tak niewinnie i platonicznie piękna, jak naszym wielkim przodkom się wydawało.

Stety czy niestety, w XX wieku matematycy nauczyli się teorię liczb wykorzystywać do bardzo przyziemnych celów.

Algorytmiczna teoria liczb

Jedną z ważnych klas problemów teorii liczb są problemy obliczeniowe. Już w czwartej klasie szkoły podstawowej dzieci uczą się, jak pisemnie dodawać dwie liczby zapisane w systemie dziesiętnym. Łatwo zauważyć, że podany na lekcjach algorytm będzie efektywny (dający się wykonać w rozsądnym czasie na kartce) dla bardzo dużych, nawet kilkudziesięciocyfrowych, liczb.

Nie każdy problem z teorii liczb umiemy efektywnie rozwiązywać. Czasami nawet dla pozornie bardzo podobnych do siebie problemów znamy bardzo różne (w sensie efektywności) jakościowo rozwiązania. Na przykład szukanie NWD dla dwóch nawet bardzo dużych liczb jest szybkie, ale już rozkład na czynniki dużych liczb – obliczeniowo jest w zasadzie poza zasięgiem najszybszych współczesnych komputerów.

Poddziedzina teorii liczb zajmująca się takimi zagadnieniami (co umiemy, a czego nie umiemy efektywnie obliczać dla dużych liczb) nazywana jest algorytmiczną teorią liczb. To właśnie głównie wyniki (zarówno negatywne, jak i pozytywne!) z tego obszaru stały się przyczynkiem do rozwoju kryptologii – jakże praktycznej gałęzi współczesnej informatyki.

Kryptologia

W twierdzeniach kryptologicznych zwykle postulujemy, że przeciwnik (podsłuchiwacz, włamywacz) – nawet jeśli ma częściowy dostęp do systemu (może podsłuchiwać to, co „w eterze”, ma ograniczony dostęp do komputera ofiary itp.) – to i tak nie jest w stanie *czegoś* zrobić – odtworzyć wiadomości, zmienić wartości jakiejś zmiennej itp. Jak w ogóle można podejść do rozwiązania tak postawionego problemu?

Rozwiązanie zadania F 988.
W chwili początkowej prędkość względna powierzchni kółek w punkcie ich styku wynosiła $\Omega_0 R$. Na powierzchnie kółek działały więc siły tarcia kinetycznego równe co do wielkości, ale skierowane przeciwnie. Większe kółko było „hamowane”, a mniejsze „przyspieszane”, aż do chwili, gdy prędkości liniowe powierzchni w miejscu ich styku wyrównały się, co oznacza, że końcowe prędkości kątowe większego kółka Ω_k i mniejszego ω_k spełniły warunek:

$$(*) \qquad \Omega_k R = -\omega_k r$$

(znak minus, bo obroty są w przeciwne strony). Niech N oznacza siłę dociskającą powierzchnie kółek, I moment bezwładności większego z nich, a i mniejszego. Mamy równania ruchu obrotowego walców:

$$IE = -fNR, \quad i\varepsilon = -fNr,$$

gdzie E i ε oznaczają odpowiednio przyspieszenia kątowe większego i mniejszego kółka (w obu równaniach występuje po prawej stronie znak minus, bo i siły, i wektory położenia punktów przyłożenia sił względem osi kółek są skierowane przeciwnie). Z upływem czasu t prędkość kątowa większego kółka Ω zmienia się jak $\Omega = \Omega_0 - tfNR/I$, a mniejszego jak $\omega = -tfNr/i$. Podstawienie tych zależności do warunku $(*)$ pozwala znaleźć czas ruchu t_k do chwili ustania poślizgu powierzchni:

$$t_k = \frac{\Omega_0 R}{fN(R^2/I + r^2/i)},$$

a następnie końcową prędkość kątową Ω_k:

$$\Omega_k = \Omega_0\left(1 - \frac{R^2}{R^2 + r^2\frac{I}{i}}\right) = \Omega_0 \frac{R^2}{R^2 + r^2}.$$

Ostatnia równość wynika z faktu, że jeśli kółka wycięto z tego samego arkusza blachy, to ich masy pozostają w stosunku jak kwadraty ich promieni, a momenty bezwładności jak czwarte potęgi promieni. Jak widać, siła tarcia (fN) występuje tylko we wzorze na t_k, co oznacza, że tę samą wartość Ω_k otrzymalibyśmy, gdyby uzgodnienie prędkości obrotu następowało natychmiast po zetknięciu kółek, tak jak można by przyjąć dla kół zębatych.

Szkic argumentacji jest zwykle podobny: wykazujemy, że gdyby przeciwnik był w stanie coś niecnego zrobić, to korzystając (być może nietrywialnie) ze szczegółów jego ataku, moglibyśmy efektywnie rozwiązać jakiś standardowy problem z algorytmicznej teorii liczb, o którym wiemy (a częściej: w który wierzymy), że jest trudny. Takie wnioskowanie sugeruje, że przesłanka była fałszywa, a więc że jednak taki sprytny przeciwnik po prostu nie istnieje.

Taher Elgamal – egipski informatyk, który w 1985 (jako pracownik firmy Hewlett-Packard) zaproponował wykorzystanie trudności problemu logarytmu dyskretnego w kryptografii klucza publicznego. Wcześniej ten problem był wykorzystywany w kontekście szyfrowania symetrycznego, co zaproponował Martin Hellman, zresztą promotor doktoratu Elgamala na Stanfordzie.

Szyfr ElGamala to jeden z dwóch najpopularniejszych (obok RSA) szyfrów kryptografii asymetrycznej. Trudność RSA wynika z trudności rozłożenia dużej liczby na czynniki pierwsze, natomiast trudność szyfru ElGamala z trudności obliczenia logarytmu dyskretnego modulo duża liczba.

I tak: szyfrowanie ElGamala da się *zredukować* do problemu trudności logarytmu dyskretnego. Problem ten polega na znalezieniu (dla danych $a, b \in \mathbb{N}$ oraz $p \in \mathbb{P}$) takiej liczby $x \in \mathbb{N}$, że

$$a^x = b \pmod{p}.$$

Wierzymy, że powyższy problem jest bardzo trudny dla dużych a, b, p. Z drugiej strony, potrafimy ściśle udowodnić, że mając w ręku algorytm A łamiący (w pewnym ściśle określonym sensie) schemat szyfrujący ElGamala, możemy skonstruować algorytm A' efektywnie rozwiązujący problem logarytmu dyskretnego.

Z powyższych dwóch faktów (a raczej: jednego postulatu „na wiarę” i jednego ścisłego rozumowania) wprost wynika, że nie może istnieć żaden algorytm łamiący szyfr ElGamala.

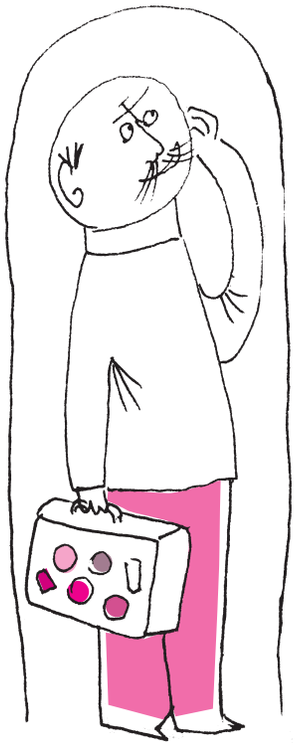
W podobnym duchu wykorzystuje się przeróżne założenia z algorytmicznej teorii liczb do dowodzenia bezpieczeństwa różnych, często bardzo nietrywialnych, protokołów kryptologicznych. Oczywiście staramy się, aby założeń teorioliczbowych było jak najmniej oraz aby były one jak najbardziej standardowe. Poza wyżej wyeksponowanym problemem logarytmu dyskretnego często wykorzystywanymi w kryptologii założeniami są m.in.:

- trudność rozkładu dużych liczb na czynniki pierwsze;
- trudność logarytmu dyskretnego w grupie punktów krzywej eliptycznej (zob. Δ^8_{18});
- trudność stwierdzania, czy dana liczba k jest resztą kwadratową modulo $n = p \cdot q$, dla pewnych danych $p, q \in \mathbb{P}$ (a jest resztą kwadratową modulo b, gdy istnieje $x \in \mathbb{N}$ taki, że $a = x^2 \pmod{b}$). Ten problem jest problemem decyzyjnym (k albo jest, albo nie jest resztą kwadratową modulo n), więc trudność oznacza tu, że nie umiemy znaleźć algorytmu, który osiąga prawdopodobieństwo sukcesu istotnie większe niż 50%;
- pewna ustalona funkcja f (np. funkcja SHA-3) ma własność funkcji jednokierunkowej, a więc dla danego losowego y bardzo trudne jest znalezienie dowolnego x takiego, że $f(x) = y$.

Powyższe założenia nie wyczerpują pełnego zakresu popularnych założeń teorioliczbowych, ale stanowią już potężną bazę, z której da się zbudować niezwykle wyrafinowane konstrukcje kryptologiczne, takie jak podpis cyfrowy, obliczenia wielopodmiotowe, dowody z wiedzą zerową, szyfrowanie asymetryczne, elektroniczna gotówka i inne.

Protokoły kryptologiczne są często bardzo pomysłowe, a dowody ich redukcji należą do trudnych zagadnień z algorytmicznej teorii liczb – niejednokrotnie są trikowe i nieoczywiste. Z braku miejsca w tym tekście nie umieszczamy konkretnych przykładów, pozostając w obszarze dywagacji ogólnych. Chętnych do dokładniejszego zgłębienia tych zagadnień zapraszam do prześledzenia cyklu *A jednak się da*, który proponowaliśmy w *Delcie* od numeru Δ^{10}_{18} do numeru Δ^8_{19}. Dodatkowo – można to zrobić, wyczulając się szczególnie na precyzyjną analizę, jakie konkretnie założenia teorioliczbowe stoją za rozpatrywanymi protokołami. To może być bardzo pouczające ćwiczenie!

Myślę, że naprawdę warto, tym bardziej że jest to przecież ta część osiągnięć ludzkości, która nawet nie śniła się takim tuzom matematyki, jak sam książę Carl Gauss.



Rozwiązanie zadania M 1620.
Odpowiedź: $n = 2$ oraz $n = 4$.

Spośród ośmiu rozważanych sum największa jest z jednej strony nie mniejsza od $8n$, a z drugiej – nie większa od $13 + 14 + 15 + 16 = 58$. Stąd wniosek, że $n \leqslant 7$. Ponadto suma ośmiu rozważanych sum jest równa dwukrotności sumy wszystkich liczb wpisanych w pola tablicy, czyli $16 \cdot 17$. Liczba n jest dzielnikiem tej sumy, wobec czego $n = 2$ lub $n = 4$. Poniższy przykład pozwala stwierdzić, że te wartości n w istocie są osiągalne.

$$\begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 9 & 11 & 13 & 15 \\ 16 & 14 & 12 & 10 \end{bmatrix}$$