Algorytmy i złożoność obliczeniowa

Damian Niwiński

Każdy adept matematyki zna momenty olśnienia rozwiązaniem trudnego problemu. Co to jednak znaczy, że problem był trudny? Zapewne nie bez znaczenia jest wiedza adepta: problem trudny dla ucznia, na przykład obliczenie długości elipsy, może nie być takim dla studenta matematyki, który dostrzega go jako przypadek szerszego zagadnienia, dla którego zna metodę rozwiązywania. Podobnie, matematyk uzbrojony w teorię grup (a najlepiej także w komputer) może potraktować jako rutynowe kolorowe zagadki Rubika. Metodę rozwiązywania zagadnień matematycznych, jaka da się zastosować w wielu (zwykle nieskończenie wielu) przypadkach nazywamy *algorytmem*. Algorytmy są być może najbardziej widomym rezultatem działalności matematyków: fizycy, inżynierowie, ekenomiści, oczekują od matematyki przede wszystkim metod, które w powtarzalnych sytuacjach pozwolą im obliczać potrzebne wielkości (które, oczywiście, mogą być nie tylko liczbami). Nie trzeba dodawać, że możliwość automatyzacji obliczeń przy pomocy komputera niepomiernie zwiększyła zainteresowanie algorytmami.

Czy zawsze, dla sensownie postawionego problemu, można dobrać stosowną metodę algorytmiczną lub choćby mieć nadzieję, że kiedyś taka metoda zostanie znaleziona? Odpowiedź jest negatywna, co zilustrujemy historią twz. dziesiątego problemu Hilberta.

W 1900 r., David Hilbert przedstawił Międzynarodowemu Kongresowi Matematyków w Paryżu listę 23 najważniejszych zagadnień, jakie, jego zdaniem, wiek dziewiętnasty pozostawił dwudziestemu do rozwiązania. Problem dziesiąty dotyczył znalezienia metody, która dla danego równania Diofantycznego (tj. równania z wieloma niewiadomymi, o współczynnikach wymiernych) rozstrzygałaby, czy istnieje rozwiązanie w liczbach całkowitych. Oczywiście, jeśli takie rozwiązanie istnieje, to zawsze w końcu można je znaleźć metodą kolejnych prób; z drugiej strony dla wielu poszczególnych równań istnieją dowody braku całkowitych rozwiązań. Jednak, jak dowiódł w 1970 r. rosyjski matematyk Jurij Matiasiewicz (wówczas 24–letni), nie istnieje algorytm, który, przyjmując jako daną równanie Diofantyczne, odpowiadałby w skończonym czasie na interesujące nas pytanie. Zauważmy, że już samo sformułowanie tego rezultatu wymagało ścisłego określenia pojęcia algorytmu. Czytelnicy artykułu W. Marka i J. Mycielskiego (Delta 12/1999) pamiętają, że dokonało się to w latach trzydziestych naszego wieku za sprawą logików: Gödla, Turinga, Posta i Churcha i Kleene'go; wtedy też opisano pierwsze problemy algorytmicznie nierozstrzygalne, tj. nie rozwiązywalne przez żaden algorytm.

Problemy nierozstrzygalne nie są jakąś rzadką anomalią w świecie matematyki; z grubsza mówiąc, nierozstrzygalność pojawia się zawsze, ilekroć problem jest na tyle ogólny, by można w nim odzwierciedlić – być może poprzez zmyślne zakodowanie – informację o wszystkich potencjalnie możliwych algorytmach. Jednak znakomita liczba zagadnień ważnych praktycznie nie jest aż tak ogólna; na przykład w pewnym zastosowaniu możemy potrzebować jedynie równań Diofantycznych o trzech niewiadomych i stopniu co najwyżej pięć.

O ile algorytm rozwiązujący problem istnieje, możliwe jest, jak wiemy, przełożenie go na program komputerowy. Można by więc pomyśleć, że granica pomiędzy problemami rozstrzygalnymi a nierozstrzygalnymi jest jednocześnie granicą stosowalności informatyki: każdy problem rozstrzygalny może być w praktyce rozwiązany przez komputer. Tak jednak nie jest. Może sie bowiem okazać, że nasz komputer już dla niewielkich danych potrzebuje takiego czasu (np. setek lat), że oczekiwanie na wynik traci sens. Dzieje się tak w szczególności, kiedy algorytm wymaga przeglądania wszystkich permutacji, lub choćby wszystkich podzbiorów jakiegoś zbioru. Nie wnikając w naturę elementarnych operacji algorytmu, możemy łatwo obliczyć, że gdyby nawet czas wykonywania takiej operacji był najmniejszym sensownym czasem fizycznym (jeden chronom, 10^{-43} sek.), to wykonanie kolejno 2^n takich operacji dla $n = 200$ przekroczyłoby czas życia człowieka, a dla $n = 500$ wiek wszechświata. Zapewne niektóre operacje mogłyby być wykonywane jednocześnie na wielu komputerach, tu jednak prędko napotkamy na nieprzekraczalne ograniczenia przestrzeni, w której takie komputery mogłyby się pomieścić (nie wspominając o kosztach).

Rozważmy dla przykładu ważne praktycznie zagadnienie znajdowania w grafie tzw. *cyklu Hamiltona*, tj. pętli, w której każdy wierzchołek występuje dokładnie raz. (Intuicyjnie, jest to najbardziej ekonomiczny sposób obejścia całego grafu.) Oczywista metoda polegałaby na przeszukiwaniu wszystkich możliwych permutacji zbioru wierzchołków grafu. Z matematycznego punktu widzenia jest to niewątpliwie algorytm, jednak jego praktyczna stosowalność ogranicza się do bardzo małych grafów.

Dla porównania, analogiczne z pozoru pytanie o istnienie tzw. *cyklu Eulera*, tj. pętli, która dokładnie raz odwiedza każdą krawędź grafu, posiada dobre rozwiązanie algorytmicznie: Czytelnik słyszał zapewne o twierdzeniu, że cykl Eulera istnieje w grafie wtedy i tylko wtedy, gdy liczba krawędzi wychodząca z każdego wierzchołka jest parzysta, ten ostatni warunek można oczywiście sprawdzić w czasie proporcjonalnym do liczby krawędzi grafu. Wspomniane rozwiązanie wymagało jednak odkrycia eleganckiej własności charakteryzującej grafy Eulera. Być może jakaś nieznana własność grafów Hamiltona pozwoliłaby i tutaj na skonstruowanie sprytnego algorytmu, działającego, powiedzmy, przynajmniej w czasie $\mathcal{O}(n^5)$? Być może, jednakże, pomimo wysiłków, żadna taka własność nie została dotąd znaleziona. Z drugiej strony, co bardzo intrygujące, nie znamy również dowodu, który wykluczyłby istnienie algorytmu rozstrzygającego interesujący nas problem w czasie $\mathcal{O}(n)$. Zagadnienie cyklu Hamiltona nie jest odosobnionym przypadkiem; w istocie znane są dziesiątki matematycznie naturalnych i praktycznie ważnych problemów, których stopień trudności obliczeniowej pozostaje nieznany.

Nie znaczy to, że nigdy nie potrafimy dowieść, że jakiś rozstrzygalny problem jest obliczeniowo trudny. Owszem, dla każdej ,,w miarę porządnej" funkcji $f(n)$ (np. dla funkcji n^2, 2^n, $n!$) potrafimy skonstruować problem nierozwiązywalny przez żaden algorytm w czasie proporcjonalnym do $f(n)$, a zarazem rozwiązywalny przez pewien algorytm pracujący w czasie $(f(n))^2$. Jednakże konstrukcja ta, wzorowana na przekątniowej konstrukcji problemu nierozstrzygalnego, jest dosyć sztuczna i rzuca niewiele światła na zagadnienie złożoności ,,prawdziwych" problemów obliczeniowych.

Przedstawiona sytuacja nie powinna nas specjalnie martwić. Jak zwykle w nauce, rzeczywiste trudności są motorem rozwoju. Współczesne studia nad algorytmami rozwijają się w dwóch kierunkach. *Teoria złożoności obliczeniowej* próbuje wyjaśniać dlaczego niektóre problemy nie poddają się próbom znalezienia dobrych rozwiązań algorytmicznych. *Algorytmika* natomiast, inspirowana potrzebami praktycznymi, niestrudzenie poszukuje takich rozwiązań, często na drodze rozszerzenia samego pojęcia algorytmu.

Ujmując rzecz pozytywnie, teoria złożoności dąży do określenia *trudności* problemów algorytmicznych i klasyfikuje je ze względu na stopień trudności. Tradycyjnie, problem obliczeniowy uważa się za praktycznie rozwiązywalny o ile istnieje algorytm, który dla danych rozmiaru n pracuje w czasie proporcjonalnym do n^k, dla pewnej stałej k. Problemy o tej własności tworzą klasę zwykle oznaczaną symbolem P (lub $PTIME$, z ang. *polynomial time*). Wspomnianą w artykule W. Marka i J. Mycielskiego (Delta 12/1999) klasę $NPTIME$ (lub NP) można określić poprzez rzutowania problemów z P, tj. $A \in NP$ o ile $A = \{x : (\exists y)\, R(x,y) \wedge |y| \leq |x|^k\}$, gdzie R jest relacją w klasie P, k jest stałą, a $|z|$ oznacza rozmiar z. Czytelnik może zauważyć, że problem cyklu Hamiltona jest w NP, gdyż relacja ,,permutacja π wierzchołków garfu G jest cyklem Hamiltona w G" jest oczywiście sprawdzalna w czasie wielomianowym.

Świat problemów obliczeniowych ma swoją strukturę, którą stopniowo poznajemy. Okazuje się na przykład, że wiele z pozoru różnych problemów jest w istocie bardzo do siebie podobnych, w tym sensie, że jeden można uznać za tłumaczenie drugiego. Wspomniane wyżej zagadnienie cyklu Hamiltona nie ma na pierwszy rzut oka związku z pytaniem, czy graf można pokolorować trzema kolorami tak, by końce krawędzi miały różne kolory, a to z kolei z pytaniem, czy dana formuła rachunku zdań jest spełniona przy jakimś wartościowaniu zmiennych. A jednak, istnienie szybkiego (wielomianowego) algorytmu dla któregokolwiek z tych problemów pociągałoby za sobą istnienie analogicznych algorytmów dla pozostałych, a nawet dla wszystkich problemów ze wspomnianej klasy NP. Problemy o tej własności nazywa się *zupełnymi* w NP. Pojęcie to zostało sformułowane pod koniec lat sześćdziesiątych przez S.Cooke'a i R.Karpa w USA i niezależnie przez L.Levina w ówczesnym ZSRR.

Badanie stopnia pokrewieństwa pomiędzy problemami i wyróżnianie problemów zupełnych jest jednym z przedmiotów teorii złożoności. Innym ważnym kierunkiem jest porównywanie różnych miar złożoności. Oprócz czasu sensownie jest bowiem badać także rozmiar pamięci operacyjnej komputera potrzebnej do rozwiązania zadania, a także np. liczbę komputerów (lub procesorów), które mogłyby rozwiązać to zadanie pracując równolegle, co może (choć nie zawsze musi) znakomicie przyśpieszyć realizację algorytmu. W ostatnich latach, w związku z rozwojem programów interakcyjnych, popularne staje się modelowanie działania komputera jako gry ze środowiskiem; implikuje to nowe miary złożoności, jak liczba rund lub liczba uczestników gry.

Przejdziemy teraz do algorytmiki, która, będąc po części dyscypliną inżynierską, próbuje jakoś radzić sobie z problemami trudnymi obliczeniowo. Jedną z możliowści jest rozwijanie algorytmów heurystycznych, działających szybko choćby w niektórych przypadkach, a także aproksymacyjnych, tj. poszukujących rozwiązań bliskich optymalnym (na przykład, zamiast cyklu Hamiltona, zadowalamy się ścieżką odwiedzającą co najmniej 95 % wierzchołków). Nie próbując wyczerpywać ogromnego tematu, wspomnimy tu o kierunkach, które nie rezygnują z walki o optymalne rozwiązania.

Jak widzieliśmy na przykładzie problemu grafów Hamiltona, ,,kamieniem filozoficznym" algorytmiki byłby sposób na przeskoczenie konieczności dokonywania wyczerpujących przeszukowań wszystkich możliwości. Oto kilka pomysłów, jak można by to uzyskać.

W tak zwanych *algorytmach probabilistycznych*, przeszukiwanie zostaje zastąpione przez losowy wybór. Ideę takiego algorytmu porównać można do sytuacji egzaminu, kiedy student wybiera, powiedzmy, 5 spośród ogłoszonych wcześniej 100 pytań (zamiast odpowiadać na wszystkie 100). Jeśli wszystkie odpowiedzi będą dobre, egzamin przyjmuje z dużym stopniem pewności, że student zna cały materiał – pewność ta w istotny sposób opiera się na fakcie, że student nie mógł przewidzieć wybranych losowo numerów pytań. Przełomowe znaczenie w algorytmice miał zaproponowany w 1976 r. przez M. O. Rabina (korzystający z idei G. Millera) probabilistyczny algorytm rozstrzygający, czy dana liczba n jest pierwsza czy złożona w czasie proporcjonalnym do $(\log n)^3$. W teście tym poszukuje się drogą losowania — nie ewentualnych dzielników liczby n, bo te mogą być rzadkie, ale – subtelnych a zarazem dość licznych ,,świadectw złożoności". Chodzi tu o liczby x spełniające alternatywę: x jest nietrywialnym pierwisatkiem z jedności mod$n$ lub też $x^{n-1} \not\equiv 1 \pmod{n}$ (kiedy n jest liczbą pierwszą, wiemy z tzw. małego twierdzenia Fermata, że nie ma takich x). Algorytm Millera–Rabina może co prawda z niewielkim prawdopodobieństwem nie wykryć żadnego świadectwa złożoności liczby złożonej i tym samym uznać ją za pierwszą, mimo to jednak, z uwagi na szybki czas i brak konkurencji, używa się go w praktyce do generowania wielkich liczb pierwszych, jakie wykorzystywane są następnie w systemie kryptograficznym RSA. Tak więc, w algorytmach probabilistycznych poświęcamy nieco pewności na rzecz szybkości.

Radykalnym rozwinięciem idei algorytmu probabilistycznego jest głośny ostatnio pomysł wykorzystania w obliczeniach efektów kwantowych. Element losowości pojawia się tu w momencie *pomiaru*, kiedy to stan kwantowy zostaje zaobserwowany jako klasyczny i informacja zawarta w tym ostatnim uznawana jest za wynik obliczenia. Zanim jednak nastąpi pomiar, komputer zachowuje się zgodnie z prawami mechaniki kwantowej. W rezultacie, n rejestrów, które w klasycznym komputerze przyjmować może w danej chwili co najwyżej jeden z 2^n możliwych ciągów n bitów $w \in \{0,1\}^n$, w komputerze kwantowym może w pewnym sensie przyjąć je wszystkie jednocześnie (na przykład, w stanie kwantowym $\frac{1}{(\sqrt{2})^n} \sum_{w\in\{0,1\}^n} [w\rangle$). Wyczerpujące przeszukiwanie może więc zostać zastąpione wygenerowaniem stanu kwantowego obejmującego naraz wszystkie możliwości.

Najbardziej jak dotąd spektakularnym sukcesem w tym kierunku jest zaproponowany przez P. Shora kwantowy algorytm rozkładu liczby całkowitej na czynniki pierwsze. Najlepszy znany ,,klasyczny" algorytm rozwiązuje to zagadnienie w czasie wykładniczym ze względu na rozmiar przedstawienia binarnego liczby n, choć z drugiej strony, faktoryzacja nie ma charakteru problemu *NP*–zupełnego. Kwantowy algorytm Shora działa w czasie wielomianowym (dokładnie, $C \cdot (\log n)^2 \cdot (\log\log n) \cdot (\log\log\log n)$). Warto wspomnieć, że możliwość szybkiej faktoryzacji dużych liczb wywołałaby spore zamieszanie, gdyż powszechnie stosowany algorytm szyfrowania RSA opiera się na hipotezie trudności tego problemu. Trzeba jednak pamiętać, że głęboki matematycznie algorytm Shora istnieje na razie jedynie na papierze, a komputery kwantowe (w odróżnieniu od kwantowej kryptografii) wydają się jeszcze dalekie od fizycznej realizacji.

Fizycznie zrealizowany został natomiast pomysł Adlemana, by do obliczeń wykorzystać reakcje biologiczne, a konkretnie zjawisko łączenia się komplementarnych odcinków DNA w słynną podwójną helisę. Jako zadanie obliczeniowe wybrano zanany nam już problem cyklu Hamiltona. W wyniku siedmiodniowego eksperymentu pojawiło się rozwiązanie. Co prawda testowany graf był niewielki, a wspomniane rozwiązanie jest dostrzegane przez człowieka w ciągu kilkudziesięciu sekund, jednak zastosowana tu technika może dla dużych grafów okazać się szybsza niż komputer cyfrowy. Pamiętajmy, że możliwość szybkiego rozwiązywania, choćby dziwaczną techniką, zagadnienia cyklu Hamiltona oznacza szybki algorytm dla dowolnego problemu klasy *NP* (tłumaczenie takiego problemu na problem cyklu Hamiltona mogłoby odbywać się przy pomocy komputera tradycyjnego). W odróżnieniu od wyrafinowanego algorytmu Shora, idea testu Adlemana jest prosta. Wierzchołki i krawędzie badanego grafu reprezentowane są przez losowo wybrane odcinki DNA w taki sposób, by wskutek komplementarności odpowiednich fragmentów, powstające w wyniku łączenia podwójne helisy reprezentowały ścieżki w grafie (nie tylko, oczywiście, ścieżki Hamiltona). Kluczowym momentem jest tu możliwość wygenerowania wszystkich lub niemal wszystkich ścieżek ,,w jednej chwili" (w ciągu kilku sekund); operacja taka jest jak wiemy praktycznie niewykonalna na klasycznym komputerze. Faza eliminacji ,,złych kandydatów", tak by w końcu pozostało jedynie poprawne rozwiązanie (tj. helisy reprezentujące ścieżki Hamiltona) wykorzystywała pomysłowo kilka różnych reakcji biologicznych i to ona była odpowiedzialna za stosunkowo długi czas. Eksperyment Adlemana otwiera nową drogę w informatyce: być może, zamiast konstruować wciąż nowe komputery wystarczy umiejętnie interpretować procesy przetwarzania informacji, jakich pełen jest świat, zwłaszcza świat materii żywej.

Wskazówki do dalszej lektury. Kompendium współczesnej wiedzy o algorytmach, podanej w atrakcyjny i przystępny sposób znajdzie Czytelnik w dostępnej w polskim przekładzie książce Th.C. Cormena, Ch.E. Leisersona i R.L. Rivesta *Wprowadzenie do algorytmów*, WNT 1997. Podstawy teorii złożoności wyłożone są w klasycznym podręczniku J.E. Hopcrofta i J.D. Ullmana *Wprowadzenie do teorii automatów, języków i obliczeń*, PWN, Warszawa 1994. Szeroką perspektywę problemów i kierunków badawczych tej teorii zakreśla monografia Ch.H. Papadimitriou *Computational complexity*, Addison–Wesley, 1995. Czytelnik zainteresowany dziesiątym problemem Hilberta może sięgnąć po książkę samego J. Matiasiewicza, Diesiataja problema Gilberta, Nauka Publishers, 1993 (przekład angielski: Yuri V. Matiyasevich, *Hilbert's Tenth Problem*, The MIT Press, 1993). Twierdzenie to przedstawione jest także w akademickim podręczniku Z. Adamowicz i P. Zbierskiego *Logika matematyczna*. PWN, Warszawa 1991. Eksperyment Adlemana przedstawiony jest przez samego autora w miesięczniku *Świat nauki*, nr 10 (86), październik 1998. Wreszcie, ciekawe spojrzenie na granice komputerowej obliczalności przedstawia fizyk R. Penrose w książkach *Nowy umysł cesarza*, PWN, Warszawa 1995 i *Makroświat, mikroświat i ludzki umysł*, Prószyński i S-ka, Warszawa 1997.