Jarosław Grytczuk

Wydział Matematyki i Informatyki
Uniwersytet Jagielloński

Czy wszystko można obliczyć?
Łagodne wprowadzenie do złożoności obliczeniowej

Frank Wilczek, laureat Nagrody Nobla z fizyki z roku 2004, został kiedyś zapytany: „Jeśli mógłbyś zadać jedno fundamentalne pytanie jakiejś nadprzyrodzonej, superinteligentnej istocie, to jak by ono brzmiało?". Jego odpowiedź zapewne zaskoczyła rozmówcę: „Czy P = NP? To pytanie zawiera w sobie wszystkie inne pytania, czyż nie?".

Niniejszy artykuł opowiada o jednym z siedmiu problemów milenijnych – zagadnieniu **Czy P = NP?**, stanowiącym jedno z największych wyzwań współczesnej nauki. Pytanie to jest centralnym problemem otwartym teorii obliczeń – dziedziny leżącej na styku matematyki i informatyki. Jednak jego znaczenie wykracza daleko poza te dyscypliny, dotykając fundamentalnych, filozoficznych pytań o naturę świata i ludzkiego umysłu. Jest przy tym rzeczą zaskakującą, że w istocie problem Czy P = NP? można sprowadzić do prostych układanek, takich jak popularne sudoku.

Czy istnieje *efektywny* sposób rozwiązania sudoku dowolnego rozmiaru? Na te i podobne pytania wciąż nie znamy odpowiedzi. Dzięki osiągnięciom teorii obliczeń wiemy jednak, że znalezienie takiego sposobu przyniosłoby zarazem rozwiązanie wszystkich podobnych problemów na świecie. Wydaje się to zatem niemożliwe, lecz dowodu matematycznego owej niemożliwości jak dotąd nie znaleziono.

1. Sudoku i inne układanki

Historia sudoku

Początków sudoku należy szukać nie tyle w Japonii, co w średniowiecznej Arabii. To tam ówcześni matematycy badali kwadraty liczbowe, których wiersze i kolumny nie zawierają powtarzających się elementów. W XVII wieku kwadraty takie także badał wybitny matematyk Leonhard Euler nazywając je kwadratami łacińskimi.

Kwadrat łaciński jako łamigłówka pojawił się po raz pierwszy w roku 1895 we francuskiej gazecie „Echo Paryża", ale nie spotkał się z uznaniem czytelników. Był to diagram 9x9 z polami do uzupełnienia, ale bez charakterystycznego podziału na 9 sektorów. W obecnej postaci łamigłówka pojawiła się po raz pierwszy w roku 1979 w amerykańskiej gazecie „Dell Magazine". Kilka lat później zawędrowała do Japonii, gdzie nadano jej obecną nazwę. W Polsce sudoku pojawiło się po raz pierwszy w roku 1996 w krzyżówkowym dodatku do „Super Expresu".

Działo się to wszystko na dobrych kilka lat przed erupcją popularności krzyżówki, którą sprokurował wielki miłośnik tego typu rozrywek – prawnik z Nowej Zelandii, Wayne Gold. On to natknął się w trakcie pobytu w Japonii na niewielką książeczką o sudoku. Po kilku latach intensywnych i pasjonujących badań zaproponował tygodnikowi „The Times" opublikowanie swoich sudoku. Stało się to w roku 2004. W ciągu niespełna roku szaleństwo sudoku ogarnęło niemal cały świat. W Polsce przyczynił się do tego tygodnik „Polityka" publikując w roku 2005 dodatek z wieloma zadaniami sudoku; patrz również [8].

Sudoku to popularna japońska krzyżówka liczbowa, która zrobiła w ostatnich latach prawdziwą furorę, dostarczając umysłowej rozrywki milionom ludzi na całym świecie. Reguła zabawy jest prosta: w puste pola kwadratu 9 x 9 należy wpisać cyfry od 1 do 9 tak, aby w każdym wierszu, w każdej kolumnie, i w każdym z 9 sektorów nie powtórzyła się żadna cyfra (rys. 1). Niewielu jednak wie, że ta niewinna układanka kryje w sobie jedną z największych tajemnic nauki, od rozwiązania której zależy być może przyszłość naszej cywilizacji. Nie dziwne więc, że na śmiałka, który sprosta wyzwaniu i odkryje ową tajemnicę czeka spora nagroda wysokości 1 000 000 dolarów, ale o tym później.

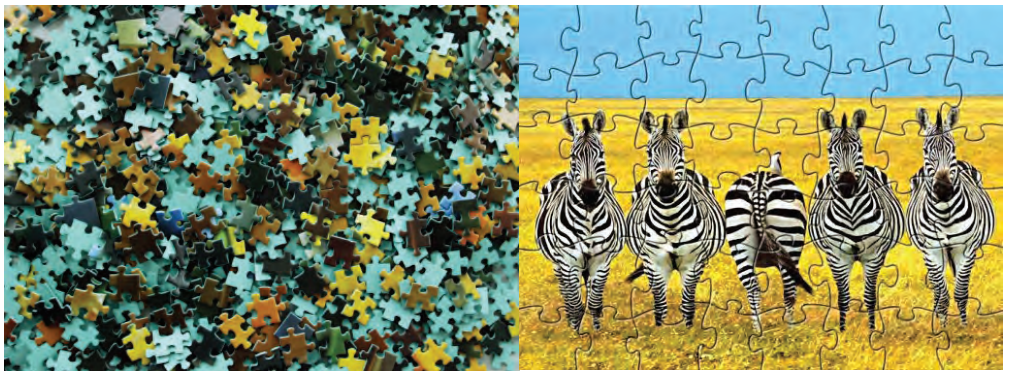
	5	4	6					
				7		9		
			8			3		6
								8
7	6							5
				5				2
9			2					
3		1					7	

1	5	4	6	9	3	2	8	7
6	3	2	5	7	8	9	4	1
8	9	7	1	2	4	6	5	3
5	1	9	8	4	7	3	2	6
2	4	3	9	6	5	7	1	8
7	6	8	3	1	2	4	9	5
4	8	6	7	5	9	1	3	2
9	7	5	2	3	1	8	6	4
3	2	1	4	8	6	5	7	9

Rysunek 1. Sudoku: przed wypełnieniem i wypełnione

Źródło: http://en.wikipedia.org/wiki/Sudoku.

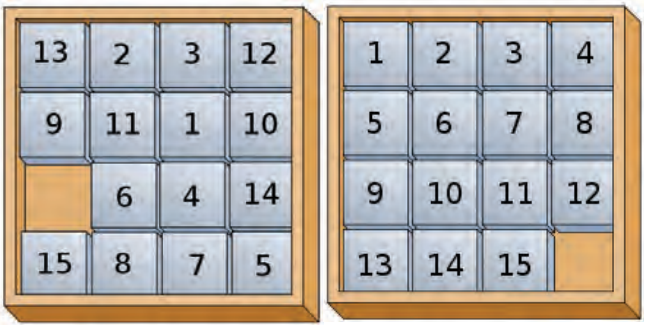
Zacznijmy od prostej obserwacji na temat natury zwykłych układanek, czyli popularnych puzzli. Kto zabawiał się układaniem obrazka z rozsypanych na stole kawałeczków, ten wie, że jest to zadanie raczej trudne, wymagające czasu, cierpliwości i koncentracji. Ale kiedy owe puzzle są już złożone, wystarczy właściwie rzut oka, by upewnić się, że wszystko się zgadza (rys. 2).



Rysunek 2. Puzzle obrazkowe

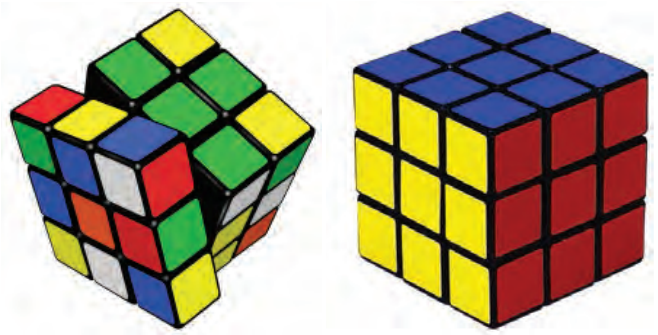
Źródło: http://www.dottysvirtualjigsaws.com/Jigs@wCreateOwnPuzzle.asp.

Podobnie rzecz się ma w przypadku nieco bardziej wyrafinowanych popularnych zabawek, jak układanka Loyda (rys. 3), kostka Rubika (rys. 4), czy też tytułowe sudoku: *sprawdzenie poprawności rozwiązania jest łatwe, niemal natychmiastowe, natomiast znalezienie owego rozwiązania jest raczej trudne, kosztuje sporo wysiłku.*



Rysunek 3. Układanka Loyda

Źródło: http://en.wikipedia.org/wiki/Fifteen_puzzle.



Rysunek 4. Kostka Rubika

Źródło: http://en.wikipedia.org/wiki/Rubik's_Cube.

Kostka Rubika

Ta popularna układanka została wynaleziona w roku 1974 przez Węgra Erno Rubika. Podobny wynalazek opatentował dwa lata później japoński inżynier Terutoshi Ishige. Podstawowa kostka ma wymiary 3x3x3 i składa się z 26 małych sześcianików zamocowanych na obrotowym przegubie. Zabawa polega na obracaniu ściankami kostki tak, aby stały się one jednobarwne. Wszystkich możliwych ustawień kostki jest aż 43 252 003 274 489 856 000. Nie przeszkadza to jednak wielu amatorom tej rozrywki w osiągnięciu imponującej szybkości rozwiązania. Obecny rekord należy do Australijczyka Feliksa Zemdegsa i wynosi 5,66 sekundy. Drugie miejsce w rankingu światowym zajmuje Polak Michał Pleskowicz z czasem 6,11 sekundy. Zawody w układaniu kostki Rubika rozgrywane są po dziś dzień na całym świecie. Istnieje także wiele odmian kostki Rubika o rozmaitych kształtach i wymiarach; patrz również [5].

Nie brakuje przykładów podobnych sytuacji również w matematyce. Znalezienie rozwiązania równania często bywa bardzo złożone, natomiast sprawdzenie jego poprawności przychodzi bez większego trudu. Poniżej równanie, którego rozwiązaniami są liczby 1, –3, 5.

$$x^3 - 3x^2 - 13x + 15 = 0$$

2. Sudoku dla komputera

To, co trudne, a nawet niewykonalne dla człowieka, bywa igraszką dla komputera. W istocie, każdą z wymienionych łamigłówek komputer rozwiąże w ułamku sekundy. Oczywiście dzieje się tak nie dlatego, że komputer jest mądrzejszy od człowieka, ale dlatego, że dysponuje nieporównanie większą mocą obliczeniową. Komputer to po prostu bardzo sprawny mechanizm, który jest w stanie sprawdzić wszystkie możliwości w bardzo krótkim czasie i w ten prymitywny, acz skuteczny sposób znaleźć rozwiązanie.

			14			8	5		10	2	4		6		
		13	10	9	11	12			6		16	15	5		
3	12	5	2	14	6		16						1	11	
	7				2		15		12		3		14	4	16
					8	7	9			6		16	3	1	5
9	1		12		15	3		4	16	7	14				13
2						13	14	8	9			12	15	7	
13	3		6	11							15		8	9	
	2	9		13							5	4		3	6
	16	10	3			9	6	7	4						1
7				8	16	5	3		15	1		9		14	12
6	8	12	1		7			3	14	11					
15	13	14		1		2		16		4				6	
	10	3						14		12	13	1	9	15	2
		2	7	4		15			3	5	10	14	16		
		11		7	3	14		9	1			8			

Rysunek 5. Sudoku 16 x 16

Źródło: http://www.sudoku.4thewww.com/other.php.

Układanka, która miałaby stanowić wyzwanie dla komputera musi mieć większy rozmiar. Wyobraźmy sobie nieco większą tabliczkę sudoku, powiedzmy o wymiarach 16 x 16 (rys. 5). Czy teraz komputer równie szybko znajdzie rozwiązanie? Na pewno nie, ale chyba czas jego poszukiwań nie zwiększy się istotnie. Być może urządzenie poda rozwiązanie po kilku czy kilkunastu sekundach, a nawet jeśli mielibyśmy poczekać parę minut, to i tak nie będzie to żaden dramat.

3. Struś Pędziwiatr

Sprawdźmy, ile czasu zajmie rozwiązanie sudoku o rozmiarach 16x16 jednemu z najszybszych komputerów na świecie, przy zastosowaniu prymitywnej metody przeszukiwania wszystkich możliwości. Komputer ten nazywa się **Struś Pędziwiatr** (ang. *Roadrunner*). Został skonstruowany w laboratoriach firmy IBM w Los Alamos. Właściwie jest to klaster (rys. 6) zajmujący powierzchnię 560 m^2, na który składa się z blisko 19 000 procesorów! W roku 2008 Struś Pędziwiatr pobił rekord świata w szybkości obliczeniowej przekraczając magiczną granicę jednego **petaflopsa**, czyli wykonując 10^{15} operacji (arytmetycznych czy binarnych) na sekundę! Dodajmy jeszcze, że kosztował on firmę IBM, bagatela, 133 000 000 dolarów.

Rysunek 6. Komputer Roadrunner

Źródło:http://en.wikipedia.org/wiki/IBM_Roadrunner.

Superkomputery

Struś Pędziwiatr królował jako najszybszy komputer świata niespełna dwa lata. W roku 2010 pokonał go Jaguar, inny amerykański superkomputer, który z kolei oddał prowadzenie na rzecz chińskiego komputera o nazwie Tianhe-1. Obecny rekord

(2012) należy do Sekwoi – również amerykańskiego komputera firmy IBM. Więcej informacji o pasjonującej rywalizacji maszyn liczących, a także o zastosowaniach ich potężnej mocy obliczeniowej, można znaleźć na stronie Top500: www.top500.org.

W naszym rachunku przyjmiemy kilka uproszczeń. Przypuśćmy, że tabliczka sudoku zawiera jedynie 25 pustych pól. W każde z nich komputer musi wpisać jedną z 4-bitowych liczb od 1 do 16. Przy pojedynczej próbie rozwiązania komputer wpisuje zatem 25 ciągów po 4 bity każdy, a więc łącznie ciąg zerojedynkowy długości 100. Wszystkich możliwości jest zatem 2^{100}, lecz tylko jedna z nich jest właściwym rozwiązaniem układanki. (W oryginalnym sudoku zdarza się, że istnieje więcej poprawnych rozwiązań, my, dla prostoty obliczeń, przyjmiemy założenie o jednoznaczności rozwiązania). Załóżmy dalej, że do sprawdzenia, czy dane uzupełnienie jest tym właściwym, potrzeba komputerowi tylko jednej operacji bitowej. Oczywiście może się zdarzyć przypadkiem, że pierwsza z brzegu ewentualność okaże się dobra, ale może być i tak, że dopiero ostatnie badane rozwiązanie jest właściwe. Zatem, w najgorszym wypadku Struś Pędziwiatr wykona

$$2^{100} = 1\ 267\ 650\ 600\ 228\ 229\ 401\ 496\ 703\ 205\ 376$$

operacji bitowych zanim znajdzie rozwiązanie. Ile czasu mu to zajmie? Hm... policzmy: 10^{15} operacji na sekundę, to daje

$$2^{100}/10^{15} = 1\ 267\ 650\ 600\ 228\ 229,\ 401\ 496\ 703\ 205\ 376 \text{ sekund.}$$

Jedna minuta ma 60 sekund, w jednej godzinie jest 60 minut, jedna doba to 24 godziny, zaś jeden rok to 365 dni. Czyli 1 267 650 600 228 229 to w przybliżeniu... 40 000 000 lat!

Oczywiście przeszukiwanie wszystkich możliwości nie jest najlepszą metodą znalezienia rozwiązania. Od czego mamy rozum, spryt i pomysłowość? Chyba istnieje jakiś sposób pozwalający rozwiązać tak niewinną układankę, jaką jest sudoku, nawet rozmiaru 100x100, w znacznie krótszym czasie. Być może taka metoda istnieje, ale jak na razie nikt jej nie znalazł. Co więcej, jeśliby się to komuś udało, to za jednym zamachem znaleziono by sposób na wszystkie układanki świata!

4. Wyścigi algorytmów

Czas wyjaśnić nieco dokładniej, o czym jest tu mowa. Przede wszystkim interesują nas **problemy obliczeniowe**, czyli takie, których rozwiązanie może

być znalezione za pomocą komputera. Do rozwiązania problemu potrzebny jest **algorytm**, w najgorszym razie przeszukujący wszystkie możliwości (to, jak widzieliśmy, może być zgoła niepraktyczne). Dla jednego problemu może istnieć wiele algorytmów różniących się pod różnymi względami, ale my skupimy się wyłącznie na porównywaniu ich szybkości.

Zabawa jest prosta i przypomina zwykłe wyścigi. Zilustrujmy to na przykładzie problemu **znajdowania największego wspólnego dzielnika** dwóch liczb. Pierwszy sposób polega na znalezieniu rozkładu na czynniki pierwsze obu liczb, a następnie wybraniu jak największej liczby wspólnych dzielników (liczby pierwsze to takie liczby naturalne, które nie dzielą się przez żadną liczbę różną od 1 i od siebie samej, przy czym jedynki do liczb pierwszych nie zaliczamy, rys. 7).

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Rysunek 7. Liczby pierwsze (w żółtych polach)

Źródło: http://www.matemaks.pl/rodzaje-liczb.php.

Jak wiadomo, każda liczba naturalna większa od 1 rozkłada się na iloczyn liczb pierwszych i to w sposób jedyny, jeśli nie zwracać uwagi na kolejność czynników. Na przykład, 84 = 2*2*3*7, zaś 234 = 2*3*3*13. Chcąc znaleźć największy wspólny dzielnik 84 i 234 wystarczy wybrać wspólne czynniki pierwsze z obu rozkładów:

$$\mathrm{NWD}(84, 234) = 2*3 = 6.$$

No tak, ale skąd wziąć rozkład na czynniki pierwsze danej liczby? Okazuje się, że z tym problemem jest podobnie jak z sudoku: nikt nie znalazł jak dotąd znacząco szybszego sposobu niźli prymitywna metoda prób i błędów.

Istnieje jednak sposób obliczenia największego wspólnego dzielnika dwóch liczb niewymagający znajdowania rozkładu na czynniki pierwsze. Jest to znany już w starożytności **algorytm Euklidesa**, polegający na sukcesywnym wykonywaniu dzielenia z resztą. Algorytm ten wykonuje z grubsza 5n operacji bitowych na danych

długości *n* nim poda wynik. Oznacza to, że z problemem rozmiaru naszej nieszczęsnej tabliczki sudoku 16 x 16 Struś Pędziwiatr poradzi sobie w ułamku sekundy![1]

Algorytm Euklidesa

Algorytm Euklidesa jest jednym z najstarszych znanych ludzkości algorytmów. Został wynaleziony przez Euklidesa około 300 lat p.n.e. Co ciekawe, pozostaje on do dziś najlepszym narzędziem znajdowania największego wspólnego dzielnika. Jest także składnikiem wielu innych algorytmów w teorii liczb, powszechnie stosowanych w rozmaitych urządzeniach elektronicznych, głównie związanych z zabezpieczeniami.

Pokażemy działanie algorytmu Euklidesa na przykładzie liczb 234 i 84. Najpierw wykonujemy dzielenie z resztą 234 przez 84:

$$234 = 2*84 + 66.$$

Następnie dzielimy 84 przez otrzymaną resztę 66:

$$84 = 1*66 + 18.$$

W kolejnych krokach dzielimy resztę otrzymaną w przedostatnim kroku przez resztę z ostatniego dzielenia:

$$66 = 3*18 + 12,$$
$$18 = 1*12 + 6,$$
$$12 = 2*6.$$

Algorytm kończy się w momencie otrzymania reszty 0. Największym wspólnym dzielnikiem początkowych liczb 234 i 84 jest ostatnia niezerowa reszta, czyli 6. Dzieje się tak dlatego, że pary liczb sąsiednich w ciągu 234, 84, 66, 18, 12, 6 mają dokładnie ten sam zbiór wspólnych dzielników.

5. Problemy łatwe

Algorytmy takie jak algorytm Euklidesa nazywamy **efektywnymi**, lub po prostu **szybkimi**. Istnieje wiele ważnych zagadnień obliczeniowych, dla których znaleziono szybkie algorytmy. To dzięki takim właśnie matematycznym wynalazkom możemy dziś błyskawicznie wyszukać w Internecie interesujące nas hasło, znaleźć najkrótszą drogę przejazdu z miasta A do miasta B, czy też bezpiecznie dokonywać przelewów bankowych.

[1] Więcej na temat algorytmu Euklidesa, w tym wyjaśnienie, dlaczego działa tak szybko, można przeczytać w rozdziale *Historia rachowania – ludzie, idee, maszyny.*

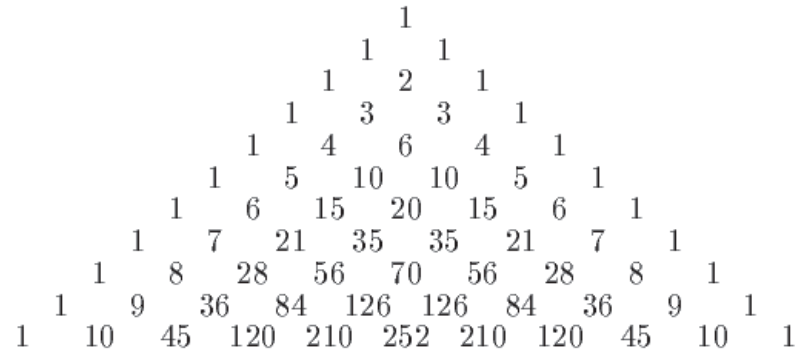
Zauważmy, że nawet jeśli liczba operacji wykonywanych przez algorytm na danych rozmiaru n wzrośnie do n^2 czy n^3, to i tak nie wpłynie to istotnie na realny czas przebiegu algorytmu (nawet dla gigantycznych danych wejściowych). W teorii obliczeń algorytm nazywamy **efektywnym** (lub **wielomianowym**), jeżeli czas jego przebiegu na danych rozmiaru n, mierzony liczbą wykonywanych operacji bitowych, jest ograniczony przez pewną potęgę n^k, gdzie k jest stałą, czyli ograniczony funkcją wielomianową zmiennej n. W żargonie takie algorytmy nazywamy **szybkimi**, a problemy obliczeniowe, które można rozwiązać za pomocą takich algorytmów – **łatwymi**. Podsumowując, problem obliczeniowy jest łatwy, jeżeli istnieje dla niego szybki algorytm.

Klasę problemów łatwych oznaczamy literą **P** (ang. *polynomial* – wielomian).

Problemami łatwymi są na przykład: znajdowanie największego wspólnego dzielnika, wyszukiwanie wzorca w tekście czy znajdowanie najkrótszej drogi w grafie.

Sprawdzanie pierwszości

Łatwy problem nie zawsze łatwo rozpoznać. Na przykład długo nie było wiadomo, czy problem sprawdzania czy dana liczba jest liczbą pierwszą jest łatwy. Dopiero w roku 2002 świat obiegła ekscytująca informacja o dokonaniu trzech Hindusów: Manindry Agrawala, Neeraja Kayala i Nitina Saxeny, którzy odkryli szybki algorytm rozwiązujący to zagadnienie. Ich artykuł noszący znamienny tytuł *PRIMES is in P*, ukazał się w prestiżowym czasopiśmie matematycznym „Annals of Mathematics". Algorytm AKS wykorzystuje nową charakteryzację liczb pierwszych podobną do tej opartej na trójkącie Pascala (rys. 9): Liczba n jest pierwsza wtedy i tylko wtedy, gdy dzieli każdą liczbę w n-tym wierszu trójkąta, za wyjątkiem skrajnych jedynek; patrz również [3].



Rysunek 8. Trójkąt Pascala

Źródło: http://en.wikipedia.org/wiki/Pascal's_triangle.

6. Kwadratura koła

Udowodnienie, że coś jest łatwe bywa czasem dość trudne. Ale jeszcze trudniej wykazać, że coś łatwym nie jest. Problem obliczeniowy nazywamy **trudnym**, jeżeli nie jest łatwy. Aby pokazać, że jakiś problem obliczeniowy jest trudny, należy zatem udowodnić, że nie istnieje szybki algorytm, który służy do jego rozwiązywania. To psychologicznie zgoła odmienna sytuacja.

Zastanówmy się dla przykładu nad problemem **faktoryzacji**, czyli rozkładu na czynniki pierwsze: daną liczbę naturalną rozłożyć na czynniki pierwsze. Na przykład:

$$2013 = 3*11*61.$$

Jak dotąd nikomu nie udało się znaleźć szybkiego algorytmu rozwiązującego to zadanie. To nie oznacza jednak samo przez się, że takiego algorytmu nie ma. Być może problem faktoryzacji jest trudny, ale żeby mieć co do tego absolutną pewność należy udowodnić, że spośród nieskończenie wielu szybkich algorytmów na świecie, żaden nie nadaje się do rozkładania na czynniki pierwsze. Tego jak dotąd również nikomu nie udało się dokonać. Nie wiadomo zatem, czy problem faktoryzacji jest łatwy, czy trudny.

Kwadratura koła

Pierwsze wzmianki o konstrukcjach kwadratu o polu przybliżającym pole danego koła znaleźć można w słynnym papirusie Rhinda z roku 1800 p.n.e. Sam problem znany był zapewne jeszcze wcześniej, w starożytnym Babilonie. W postaci klasycznej konstrukcji geometrycznej sformułowali go po raz pierwszy Pitagorejczycy. Pomimo sporych wysiłków nie udało się jednak greckim matematykom znaleźć poszukiwanej konstrukcji. Musieli oni zadowolić się rozwiązaniami przybliżonymi. Na przykład, Archimedes wpisał w koło i opisał na nim sześciokąt foremny, dziewięciokrotnie podwoił liczbę jego boków, a następnie zamienił ten wielokąt na kwadrat.

Rozstrzygnięcia problemu nie przyniosły także wieki średnie ani epoka nowożytna, choć zajmowali się nim najznakomitsi matematycy tamtych czasów – Fibonacci, François Viete, Gotfried Wilhelm Leibniz czy Leonhard Euler. Jedną z najciekawszych konstrukcji przybliżonych podał polski matematyk i mechanik Adam Kochański (1631-1700).

Ostateczne potwierdzenie narastającego przekonania o niemożliwości kwadratury koła nadeszło od strony algebry. Dzięki genialnemu pomysłowi Kartezjusza, figury geometryczne można opisywać za pomocą równań algebraicznych w prostokątnym układzie współrzędnych. Punkty, które da się skonstruować za pomocą cyrkla i linijki mają charakterystyczną postać tzw. pierwiastników (liczb powstających na bazie

ułamków poprzez wielokrotne nakładanie pierwiastka kwadratowego oraz dodawanie i mnożenie). Dzieje się tak dlatego, że prostą (linijkę) opisuje równanie pierwszego stopnia, zaś koło (cyrkiel) – równanie stopnia drugiego. Współrzędne punktów przecięcia prostych i okręgów powstające w toku konstrukcji muszą być zatem rozwiązaniami równań algebraicznych, których stopień wyraża się potęgą dwójki. Gdyby konstrukcja kwadratury koła była możliwa, to taką liczbą musiałaby być liczba π. Ale w roku 1880 Lindemann udowodnił, że π jest liczbą przestępną – w ogóle nie istnieje równanie algebraiczne, którego π jest pierwiastkiem.

To dokonanie ostatecznie zamknęło kwestię kwadratury koła w świecie nauki. O dziwo, wśród amatorów matematyki do dziś zdarzają się sceptycy poszukujący uparcie tej nieuchwytnej konstrukcji.

Sytuacja przypomina tę sprzed kilkuset lat, kiedy to borykano się ze słynnym zagadnieniem **kwadratury koła**. Problem polegał na znalezieniu geometrycznej konstrukcji (za pomocą cyrkla i linijki) kwadratu o polu równym polu danego koła. Ponieważ przez setki lat od sformułowania tego zadania nikomu takiej konstrukcji nie udało się znaleźć, większość matematyków zaczęła sądzić, że jej po prostu nie ma. W końcu udało się udowodnić, że kwadratura koła jest niemożliwa, przy użyciu metod nowoczesnej na owe czasy algebry. Być może historia powtórzy się w przypadku **problemu faktoryzacji**...

Problem faktoryzacji jest w pewnym sensie odwrotny do problemu sudoku. Mamy tu już złożoną układankę w postaci danej liczby, a szukamy puzzli, które ją tworzą, czyli liczb pierwszych, których iloczyn daje tę liczbę. Jeśli jednak ktoś dostarczy nam rozkładu, to łatwo sprawdzimy czy jest on poprawny wykonując zwykłe mnożenie. Pod tym względem problem faktoryzacji i sudoku są podobne.

W tym miejscu warto przytoczyć zabawną sytuację, która miała miejsce na kongresie matematycznym w 1904 roku. Otóż matematyk amerykański Frank Nelson Cole w czasie swojego wystąpienia podszedł do tablicy i napisał

$$2^{67} - 1 = 147\,573\,952\,589\,676\,412\,927 = 193\,707\,721 * 761\,838\,257\,287,$$

a następnie dowiódł prawdziwości napisanej równości poprzez pomnożenie liczb sposobem pisemnym. Wszystko to odbyło się w pełnej napięcia ciszy. Dopiero na koniec wyznał, że znalezienie owego rozkładu zajęło mu, bagatela, 20 niedziel!

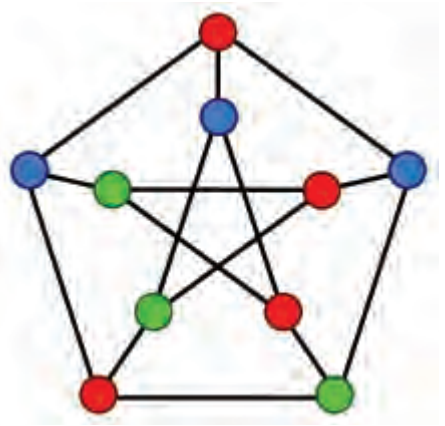
Liczby pierwsze postaci $2^n - 1$ stanowią przedmiot fascynacji matematyków od dawna (noszą one nazwę **liczb Mersenne'a**). Nietrudno wykazać, że jeżeli $2^n - 1$ jest liczbą pierwszą, to i n musi być liczbą pierwszą. Przykład liczby Cole'a pokazuje, że na odwrót nie zawsze to zachodzi. Oczywiście dzisiaj do sprawdzenia, czy $2^n - 1$ jest liczbą pierwszą użyjemy komputera. Od roku 1996 trwa w Internecie wielkie poszukiwanie liczb pierwszych Mersenne'a w ramach projektu GIMPS (http://www.mersenne.org/). Największą liczbą Mersenne'a znalezioną w ten

kolektywny sposób, i zarazem największą znaną obecnie liczbą pierwszą, jest $2^{43\,112\,609} - 1$. Do dziś nie wiadomo, czy liczb Mersenne'a jest nieskończenie wiele.

7. Klasa NP

Istnieje mnóstwo problemów obliczeniowych, o których nie wiemy, czy są łatwe, czy trudne. Najważniejszą klasę tworzą problemy o naturze układanek: łatwo zweryfikować poprawność podanego rozwiązania, natomiast nie zawsze prosto je znaleźć. Klasę tę oznaczamy symbolem **NP** (ang. *nondeterministic polynomial*). W tej klasie znajduje się zarówno problem sudoku i problem faktoryzacji, jak i całe mnóstwo kombinatorycznych zagadnień optymalizacyjnych, spośród których przedstawimy poniżej dwa, polecamy tutaj łagodne wprowadzenie do złożoności obliczeniowej [4].

Pierwsze z nich to problem **3-kolorowania grafu**: czy dany graf da się pokolorować poprawnie trzema kolorami? Kolorowanie grafu jest **poprawne,** jeżeli żadne dwa wierzchołki połączone krawędzią nie są tego samego koloru (rys. 9).



Rysunek 9. Poprawne pokolorowanie grafu

Źródło: http://en.wikipedia.org/wiki/Graph_coloring.

Poprawność pokolorowania grafu jest łatwo zweryfikować, wystarczy przejrzeć wszystkie krawędzie grafu i porównać kolory ich końców. Problem ten jest więc w klasie NP. Ale jak stwierdzić, czy poprawne 3-kolorowanie danego grafu w ogóle istnieje? Można oczywiście sprawdzić wszystkie możliwości, ale to może trwać długo, wszak jest ich aż 3^n (gdzie n oznacza liczbę wierzchołków grafu). Podobnie jak w przypadku faktoryzacji nie wiemy, czy 3-kolorwanie grafu jest łatwe, czy trudne. Nietrudno natomiast przekonać się, że analogiczny problem 2-kolorowania grafu jest łatwy.

Drugim przykładem popularnego problemu z klasy NP jest **problem SAT**, w którym pytamy się, czy dana formuła logiczna jest **spełnialna**, to znaczy, czy istnieje takie podstawienie zer i jedynek w miejsce zmiennych, że wartość logiczna danej formuły wyniesie 1. Na przykład formuła

$$F = (p \vee q \vee r) \wedge (p \vee \neg q \vee s) \wedge (q \vee r \vee \neg s)$$

jest spełnialna, ponieważ przy podstawieniu $p = q = 1$ oraz dowolnych wartościach r i s, w każdym nawiasie pojawi się jedynka:

$$F = (1 \vee 1 \vee r) \wedge (1 \vee 0 \vee s) \wedge (1 \vee r \vee \neg s).$$

Łatwo jest więc zweryfikować poprawność pojedynczego podstawienia, jednak jak dotąd nie wynaleziono szybkiego algorytmu rozstrzygającego istnienie podstawienia spełniającego, ale nie wykluczono też takiej możliwości. Ciągle nie wiadomo, czy problem SAT jest na pewno trudny.

8. Problem milenijny

Czy wobec tego w ogóle znany jest jakiś przykład problemu w klasie NP, o którym wiemy już, że na pewno nie jest łatwy? No właśnie nie! I to jest nasz główny problem. Po wielu latach zmagań i wysiłków, wciąż jesteśmy skazani na spekulacje:

Czy P = NP?

Wiele wskazuje na to, że odpowiedź na to pytanie jest negatywna. Równość P = NP oznaczałaby, że właściwie nie ma naturalnych problemów obliczeniowych, które byłyby trudne, że, cytując słowa Margaret Fuller, *wszystko jest trudne nim stanie się łatwe*. Wszystkie układanki świata da się rozwiązać szybko, wszystkie twierdzenia w matematyce da się udowodnić „mechanicznie". Równałoby się to w gruncie rzeczy ze stwierdzeniem, że ludzką kreatywność dałoby się w pewnym sensie zautomatyzować. Czy możemy wyobrazić sobie komputer komponujący muzykę tak wspaniałą jak muzyka Chopina? Chyba nie, chociaż istnieją już programy komputerowe potrafiące tworzyć muzykę naśladującą style sławnych kompozytorów.

W roku 2000 z okazji przełomu tysiącleci wskazano siedem najważniejszych problemów matematycznych oraz wyznaczono nagrody pieniężne za rozwiązanie każdego z nich, każda w wysokości 1 000 000 dolarów. Fundatorem tych nagród, jak i całego instytutu matematycznego swojego imienia jest ame-

rykański biznesmen Landon T. Clay. Problem rozstrzygnięcia czy P = NP jest jednym z owych siedmiu problemów milenijnych.

W tym miejscu warto wspomnieć, że jak dotąd rozwiązano tylko jeden z problemów milenijnych. Chodzi o słynną **hipotezę Poincarégo**, dotyczącą powierzchni 3-wymiarowych w przestrzeni 4-wymiarowej. Udowodnił ją w roku 2006 Rosjanin Grigorij Perelman. Ku zdumieniu świata, nie zechciał on odebrać swojej nagrody; patrz [7].

Grigorij Perelman (1966-)

Perelman udostępnił w Internecie manuskrypt z rozwiązaniem hipotezy Poincarégo w 2003 roku. Sprawdzenie poprawności wszystkich zawiłych rozumowań zajęło ekspertom trzy lata. W roku 2006 ogłoszono triumfalnie rozwiązanie hipotezy i przyznano Perelmanowi medal Fieldsa – jedną z najbardziej prestiżowych nagród matematycznych. Jednak Perelman odmówił jej przyjęcia, twierdząc, że doniosłość odkrycia matematycznego może być zweryfikowana dopiero po wielu latach od jego dokonania.

Odmowa przyjęcia „matematycznego Nobla" odbiła się głośnym echem, nie tylko w środowisku matematycznym. Skromny, acz ekscentryczny geniusz stał się z dnia na dzień bohaterem licznych medialnych doniesień.

Wrzawa wokół jego osoby wybuchła ponownie po odmowie przyjęcia miliona dolarów nagrody ufundowanej przez Instytut Matematyczny Claya. Tym razem Perelman jako powód wskazał niesprawiedliwe, jego zdaniem, pominięcie Richarda Hamiltona jako matematyka, który przyczynił się w równym stopniu do rozwiązania hipotezy. W istocie faktem jest, że to właśnie Hamilton zaproponował właściwe podejście do hipotezy Poincarégo, a także do ogólniejszej hipotezy Thurstona o klasyfikacji powierzchni trójwymiarowych.

Obecnie Grisza Perelman jest bezrobotnym matematykiem, mieszkającym w skromnym mieszkaniu na petersburgskim blokowisku, rzadko kontaktującym się ze światem zewnętrznym.

9. NP-zupełność

Problem P = NP? może sprawiać wrażenie zagadnienia bardzo rozległego – wszak chodzi tu o porównanie dwóch nieskończonych klas problemów obliczeniowych. Po głębszym badaniu okazało się jednak, że właściwie cała zagadka sprowadza się do pojedynczego problemu, takiego jak np. sudoku. Czyżby los ludzkości zależał od tej niewinnej układanki?

Aby wyjaśnić, w czym rzecz, rozważmy ponownie problem SAT i problem 3-kolorowania grafu. Otóż można udowodnić, że dla każdego grafu G istnieje for-

muła logiczna $F = F(G)$ taka, że G jest 3-kolorowalny wtedy i tylko wtedy, gdy F jest spełnialna. Ponadto, skonstruowanie takiej formuły „kodującej" problem 3-kolorowalności grafu jest możliwe w czasie wielomianowym. Oznacza to, że w istocie problem 3-kolorowalności grafu sprowadza się łatwo do problemu SAT. Zatem, jeżeli problem SAT jest łatwy, to problem 3-kolorowalności też jest łatwy.

W roku 1971 Kanadyjczyk Stephen Cook dowiódł, że właściwie każdy problem z klasy NP w podobny sposób sprowadza się do problemu SAT. Mówimy, że problem SAT jest **NP-zupełny**.

Jest to zadziwiające zjawisko: oto rozstrzygnięcie, czy P = NP sprowadza się w zasadzie do zbadania jednego jedynego problemu obliczeniowego – problemu SAT. Jeśli problem ten jest trudny, to oczywiście mamy P ≠ NP. Jeśli jest łatwy, to wszystkie problemy w klasie NP są łatwe i mamy P = NP. Jest jeszcze trzecia możliwość, ale o tym nieco dalej.

Odnotujmy jeszcze, iż problem SAT nie jest jedynym NP-zupełnym problemem obliczeniowym. W rzeczywistości znaleziono wiele takich problemów w kombinatoryce, a jednym z nich jest także sudoku. Dowiódł tego Japończyk Takayuki Yato w roku 2003. Rezultat ten oznacza, że dowolny problem z klasy NP (np. SAT, 3-kolorowalność, faktoryzację itp.) można „zakodować" w postaci odpowiedniej tabliczki sudoku, której rozwiązanie da pośrednio rozwiązanie danego problemu.

Problemy NP-zupełne

Problem SAT nie jest jedynym znanym problemem NP-zupełnym. Jest nim także problem 3-kolorowania grafu i wiele innych naturalnych problemów o charakterze optymalizacyjnym. Na przykład, słynny problem komiwojażera, w którym mamy graf pełny z liczbami na krawędziach, a zadanie polega na znalezieniu cyklu przechodzącego przez wszystkie wierzchołki grafu tylko raz, którego suma jest minimalna[2].

Problemy NP-zupełne to w pewnym sensie najtrudniejsze problemy w klasie NP – podanie algorytmu wielomianowego dla jednego z nich pociąga za sobą istnienie algorytmów wielomianowych dla wszystkich problemów NP-zupełnych. Ustalenie, czy dany problem jest NP-zupełny bywa czasem dość trudne. Nie wiadomo na przykład, czy problem faktoryzacji jest NP-zupełny, czy nie.

[2] W sytuacji, gdy nie jest znany wielomianowy, czyli szybki algorytm rozwiązywania problemu optymalizacyjnego, konstruowane są algorytmy, które dostarczają rozwiązania przybliżone, a więc nie zawsze najlepsze.

10. Między prawdą a nieprawdą

Wydawać by się mogło, że problem Czy P = NP? może mieć jedynie dwa rozstrzygnięcia: albo P = NP, albo P ≠ NP. Okazuje się, że istnieje jeszcze trzecia możliwość...

W roku 1938 **Kurt Gödel** dokonał odkrycia, które wstrząsnęło fundamentami matematycznego gmachu. Wykazał mianowicie, że istnieją w matematyce hipotezy, których nigdy nie da się rozstrzygnąć – ani potwierdzić, ani obalić! Od tej pory matematycy liczą się z tym, że właściwie każdy problem otwarty może okazać się **nierozstrzygalny**. Rozważmy dla przykładu słynną **hipotezę $3x + 1$**. Dotyczy ona liczbowej zabawy, którą możemy rozpocząć od dowolnej liczby naturalnej n. Jeżeli n jest parzyste, to dzielimy n przez 2. Jeżeli n jest nieparzyste, to mnożymy n przez 3 i dodajemy 1. Z nową liczbą postępujemy podobnie, z kolejną tak samo, i tak dalej. Na przykład, jeżeli $n = 7$, to w pierwszym kroku dostajemy 3*7 + 1 = 22. Ponieważ 22 jest liczbą parzystą, dzielimy 22 przez 2 i dostajemy 11. Ta ostatnia liczba jest z kolei nieparzysta, zatem obliczamy 3*11 + 1 = 34. Znów dzielimy przez 2 i dostajemy 17. W kolejnych krokach dostajemy ciąg liczb: 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, i w końcu 1. Hipoteza $3x + 1$ głosi, że zaczynając tę procedurę od dowolnej liczby naturalnej n, zawsze dojdziemy na końcu do 1. Tę zabawę można zapisać w postaci następującego algorytmu:

```
while n ≠ 1 do
    if n mod 2 = 0 then n:= n/2
    else n:= 3*n + 1
```

Jest rzeczą zadziwiającą, że tak prostej zagadki najtęższe matematyczne umysły nie potrafią rozwiązać. W czym leży trudność tego problemu? Niektórzy eksperci sądzą, że być może przyczyną jest nierozstrzygalność. Być może to, czego szukamy, czyli matematyczny dowód prawdziwości tej hipotezy, po prostu nie istnieje. Ale czy to nie oznacza, że hipoteza jest po prostu fałszywa? Że musi istnieć liczba n, z której nigdy do 1 nie dojdziemy? No właśnie, że nie! Równie dobrze może być prawdą, że i dla przeciwnej hipotezy nie ma matematycznego dowodu. Tego właśnie dowiódł Gödel: istnieją w elementarnej arytmetyce zdania, których nie da się udowodnić, ale zarazem nie da się udowodnić ich negacji. Być może takim zdaniem jest hipoteza $3x + 1$, ale tego też jak dotąd nikomu nie udało się udowodnić.

Podobnie rzecz się ma z problemem Czy P = NP?. Czyżby jego rozstrzygnięcie leżało poza zasięgiem matematycznej dedukcji?

Kurt Gödel (1906-1978)

Odkrycia Gödla zalicza się do najdonioślejszych dokonań myśli ludzkiej XX wieku. Jego największym osiągnięciem jest z pewnością wspomniane twierdzenie o niezupełności arytmetyki, gwarantujące istnienie zdań, których nie da się udowodnić ani obalić. Główny pomysł przypomina słynny paradoks kłamcy: „To zdanie jest fałszywe". Gödel wyszedł od podobnego zdania: „To zdanie nie ma dowodu", umiejętnie je matematycznie formalizując. Należy jeszcze dodać, że rezultaty te są słuszne jedynie przy założeniu niesprzeczności arytmetyki, która oznacza, że nie jest możliwe, aby jakieś twierdzenie i jego zaprzeczenie jednocześnie można było udowodnić. Jeden z największych matematyków wszechczasów David Hilbert postulował udowodnienie niesprzeczności arytmetyki. Jednak projekt ten okazał się w świetle wyników Gödla niemożliwy do realizacji.

Gödel zajmował się również teorią względności. Wykazał on m.in., że teoria Einsteina dopuszcza możliwość podróży w czasie. Niedawno odkryto również, że to on właściwie jako pierwszy, w liście do Johna von Neumanna, sformułował problem równoważny z problemem Czy P = NP?

11. Czy wszystko można obliczyć?

Dokonania Kurta Gödla w dziedzinie podstaw matematyki odegrały także znaczącą rolę w narodzinach informatyki. To właśnie one zainspirowały **Alana Turinga** do sformułowania pojęcia **maszyny Turinga** – fundamentu współczesnej teorii obliczeń stanowiącego matematycznie ścisłą formalizację pojęcia algorytmu. Szkoda, że ten wybitny matematyk nie dożył czasów, kiedy to na każdym niemal kroku natknąć się można na fizyczną realizację jego *automatic machine*.

Jedna z prostych, acz nieoczywistych rzeczy wynikająca z tej formalizacji to istnienie problemów decyzyjnych algorytmicznie nierozstrzygalnych. Cóż to jest takiego problem decyzyjny? To po prostu problem dotyczący liczb naturalnych, w którym jedyne możliwe odpowiedzi to TAK lub NIE. Na przykład: „Czy n jest liczbą pierwszą?". Jeżeli $n = 17$, to odpowiedź brzmi TAK, jeśli $n = 18$, to wówczas odpowiedź brzmi NIE. Problem decyzyjny nazywamy **rozstrzygalnym algorytmicznie**, jeżeli istnieje algorytm (w sensie Turinga), który dla zadanego n znajduje prawidłową odpowiedź – TAK lub NIE.

Zamieniając TAK na 1 i NIE na 0, widzimy, że każdy problem decyzyjny możemy uważać za nieskończony ciąg zerojedynkowy. Początek tego ciągu dla problemu pierwszości to 0110101000101... Nie wchodząc w szczegóły definicji pojęcia algorytmu zgodzimy się zapewne, że jego opis (np. w postaci programu komputerowego) to pewien **skończony** ciąg zerojedynkowy. Z kolei skończony ciąg zerojedynkowy możemy traktować jako binarne przedstawienie pewnej liczby natu-

ralnej. Widzimy zatem, że wszystkich możliwych algorytmów jest nie więcej niż liczb naturalnych. Natomiast wszystkich możliwych problemów decyzyjnych jest więcej – nie da się bowiem wszystkich **nieskończonych** ciągów zerojedynkowych ponumerować liczbami naturalnymi. W istocie, wyobraźmy sobie, że jednak to się udało i mamy listę wszystkich takich ciągów C_1, C_2, C_3,... . Pomyślmy teraz o ciągu X, którego pierwszy wyraz różni się od pierwszego wyrazu ciągu C_1, drugi wyraz różni się od drugiego wyrazu ciągu C_2, trzeci wyraz różni się od trzeciego wyrazu ciągu C_3, i tak dalej. Ciąg X jest więc ściśle określonym nieskończonym ciągiem zerojedynkowym, powinien więc znajdować się na naszej liście. Ale którym ciągiem z listy może być X? Pierwszym? Nie, bo różni się od C_1 na pierwszej pozycji. Drugim? Też nie – przecież różni się od niego na drugiej pozycji. Setnym? Nie, albowiem setny wyraz ciągu C_{100} jest inny niż setny wyraz X. Ciągu X nie ma na liście. Zatem lista taka, obejmująca wszystkie ciągi, nie może istnieć.

Powyższy eksperyment myślowy pokazuje, że istnieją problemy decyzyjne, których nie da się rozwiązać za pomocą żadnego algorytmu. Przykładu takiego problemu dostarczył jako pierwszy sam Turing w swojej fundamentalnej pracy z roku 1936 *On computable numbers, with an application to the Entscheidungsproblem*. Jest to tzw. **problem stopu** – na wejściu dostajemy algorytm A wraz z pewnymi danymi wejściowymi (np. algorytm $3x + 1$ i liczbę $n = 7$) i mamy stwierdzić, czy algorytm A zatrzyma się na tych danych. Jak wykazał Turing problem ten jest nierozstrzygalny – nie istnieje algorytm, który rozwiązuje tak postawione zagadnienie.

Problem stopu

Podamy proste uzasadnienie, że problem stopu nie jest rozstrzygalny, czyli że nie jest możliwe napisanie programu, który może zbadać każdy inny program i stwierdzić w każdym przypadku, czy po uruchomieniu zatrzyma się on, czy też wejdzie w nieskończoną pętlę.

Załóżmy jednak, że istnieje funkcja logiczna `T(R)`, której argumentem `R` jest jakikolwiek program i `T(R) = True`, jeśli program `R` kończy swoje działania, `T(R) = False`, jeśli program `R` nie kończy działania. Rozważmy teraz następujący podprogram `P`:

```
proc P;
   while T(P) do;
   return
```

Łatwo zauważyć, że jeśli `T(P) = True`, to program `P` zapętla się, a kończy działanie tylko wtedy, gdy `T(P) = False`. W obu przypadkach funkcja `T(P)` ma złą wartość i ta sprzeczność pokazuje, że funkcja `T` nie może istnieć.

Jak to jednak często bywa, o konkretnym problemie decyzyjnym nie zawsze łatwo rozstrzygnąć, czy jest rozstrzygalny, czy też nie. Jednym z najsławniejszych zagadnień tego typu był dziesiąty problem Hilberta dotyczący rozwiązalności równań diofantycznych (takich jak np. słynne równanie Fermata $x^n + y^n = z^n$). Dopiero po 70 latach Rosjanin Yuri Matiyasevich udowodnił, że problem Hilberta jest nierozstrzygalny algorytmicznie.

Alan Turing (1912-1954)

Stworzenie teoretycznego modelu współczesnego komputera to niejedyne osiągnięcie Alana Turinga, które istotnie wpłynęło na losy świata. Podczas II wojny światowej pracował w ośrodku w Bletchley Park pod Londynem w zespole brytyjskich kryptologów nad łamaniem szyfrów Enigmy, niemieckiej maszyny szyfrującej. Dzięki wcześniejszym pracom polskich kryptologów – Mariana Rejewskiego, Henryka Zygalskiego i Jerzego Różyckiego, którzy już w roku 1932 złamali kod Enigmy starszego typu – zespół Turinga skonstruował specjalne urządzenia (tzw. bomby kryptologiczne), również bazując na pomyśle polskich kryptologów, które pomagały rozszyfrowywać niemieckie depesze praktycznie przez cały okres działań wojennych od roku 1940. W Bletchley Park od roku 1943 pracował również pierwszy elektroniczny komputer Colossus.

Po wojnie Turing pracował m.in. nad budową pierwszego brytyjskiego komputera według architektury von Neumanna. W tym samym czasie rozpoczyna badania nad analogiami działania maszyny liczącej i ludzkiego mózgu, studiując w tym celu fizjologię i neurologię. W efekcie proponuje swój słynny test Turinga, jako metodę rozstrzygnięcia czy maszyna potrafi „myśleć". Test ten polega na rozmowie sędziego – człowieka w języku naturalnym z pozostałymi stronami poprzez ekran komputera. Jeśli sędzia nie jest w stanie określić, czy któraś ze stron jest maszyną czy człowiekiem, wtedy mówi się, że maszyna przeszła test. Zakłada się, że zarówno człowiek, jak i maszyna próbują przejść test zachowując się w sposób możliwie zbliżony do ludzkiego.

Od roku 1991 urządzane są zawody o nagrodę Loebnera bazujące na teście Turinga. Jak dotąd żadnemu programowi nie udało się zdobyć złotego medalu Loebnera.

12. Epilog

W opinii większości ekspertów mało prawdopodobne jest, abyśmy ujrzeli rozwiązanie problemu Czy P = NP? w najbliższej przyszłości. Nie jest to jednak powód do pesymizmu. W gruncie rzeczy obecna sytuacja sprzyja rozwojowi teorii obliczeń – nic tak bowiem nie stymuluje badań naukowych jak twardy

opór materii problemu. Chęć pokonania trudności, sprostania wyzwaniu, zaspokojenia nabrzmiałej ciekawości – to od wieków główny motor napędowy nauki. I często ważniejsze i donioślejsze okazują się od samej odpowiedzi na dręczące nas pytanie, dokonania służące jej znalezieniu. Jakiż pożytek płynie dla ludzkości, lub choćby dla samej matematyki, z faktu, że suma dwóch n-tych potęg liczb naturalnych nigdy nie jest n-tą potęgą dla $n > 2$? Nie wydaje się, aby poza doznaniem estetycznym, własność ta miała jakiekolwiek znaczenie. Jednakowoż, matematycy zrobili wiele, aby swoją pewność w tym względzie posiąść, budując przez setki lat potężny aparat matematyczny, którego wykorzystanie wykracza daleko poza elementarną arytmetykę. Niechaj więc jak najdłużej pytanie Czy P = NP? odpiera ataki ludzkiego geniuszu, wciągając nas w głąb tajemniczego świata obliczeń.

Literatura

1. Bartnicki T., *Jak wygrać milion dolarów w Sapera*, „Matematyka-Społeczeństwo-Nauczanie" 2008, nr 40
2. Conway J.H., Guy R.K., *Księga liczb*, WNT, Warszaw 2006
3. Czerwiński W., *Test na liczbę pierwszą*, „Delta" czerwiec 2012
4. Harel D., *Rzecz o istocie informatyki*, WNT, Warszawa 2001
5. Hintze W., *Magiczna kostka*, PWN, Warszawa 1987
6. Kordos M., *Wykłady z historii matematyki*, WSiP, Warszawa 1994
7. Strzelecki P., *Hipoteza Poincarego*, „Delta" styczeń 2004
8. Wilson R.J., *Jak rozwiązywać sudoku*, Dom Wydawniczy Rebis, Poznań 2005
9. Yan S.Y., *Teoria liczb w informatyce*, PWN, Warszawa 2006

Prof. nzw. dr hab. Jarosław Grytczuk

ukończył studia matematyczne w Wyższej Szkole Pedagogicznej w Zielonej Górze (obecnie Uniwersytet Zielonogórski). Doktorat z matematyki został mu nadany w roku 1996 na Uniwersytecie Adama Mickiewicza w Poznaniu. Stopień doktora habilitowanego uzyskał w roku 2006, również na UAM. Obecnie Jarosław Grytczuk jest profesorem nadzwyczajnym na Wydziale Matematyki i Informatyki Uniwersytetu Jagiellońskiego oraz na Wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej. Stale współpracuje z czołowymi przedstawicielami matematyki dyskretnej i informatyki teoretycznej zarówno w kraju, jak i zagranicą. Jest współorganizatorem Polskich Konferencji Kombinatorycznych odbywających się cyklicznie od roku 2006. Prowadzi również działalność popularyzatorską, występując często na wykładach dla młodzieży. W swoim dorobku naukowym posiada około 40 prac opublikowanych w czasopismach o zasięgu międzynarodowym.

grytczuk@tcs.uj.edu.pl
http://tcs.uj.edu.pl/Grytczuk