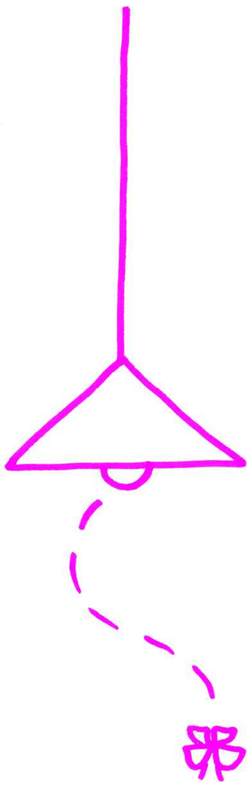
Na granicy możliwości

Wojciech CZERWIŃSKI[*]



Algorytm to sposób rozwiązania pewnego problemu. Informatyka i matematyka od dawna badają różnego rodzaju problemy, szukając dla nich algorytmów, najczęściej możliwie szybkich. My jednak tym razem postąpimy wręcz przeciwnie: zajmiemy się algorytmami wyjątkowo wolnymi.

Najpierw wypada powiedzieć, że istnieją problemy, dla których w ogóle nie istnieje algorytm, który je rozwiązuje. Nie tylko nie znamy ani jednego, ale nawet potrafimy udowodnić, że żadnego nie ma. Takie problemy nazywa się nierozstrzygalnymi; najsławniejszy to *problem stopu.* Formułuje się go tak: dany jest pewien algorytm i należy stwierdzić, czy program, który będzie go realizował, kiedyś się zatrzyma. Nierozstrzygalność tego problemu oznacza więc, że nie istnieje żaden ogólny sposób, żeby stwierdzić, czy program, któremu właśnie się przyglądamy, zatrzyma się w pewnym momencie czy też będzie pracował w nieskończoność.

Tym razem zajmiemy się jednak metodą konstruowania algorytmów dla problemów, które co prawda można rozwiązać, ale nie istnieje dla nich żaden algorytm działający w jakimkolwiek rozsądnym czasie. O takich problemach mówimy, że nie są pierwotnie rekurencyjne. Można śmiało stwierdzić, że są one prawie nierozstrzygalne – na granicy możliwości rozwiązania. Żeby lepiej wyobrazić sobie, jak dziwne muszą być takie zagadnienia, sprecyzujmy, co rozumiemy przez problem, który nie jest pierwotnie rekurencyjny. Powiedzmy, że dane wejściowe są wielkości n, jeśli mają n bitów – w tym sensie, na przykład, liczba

$$2014 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1 = \langle 11111011110 \rangle_2$$

da się zapisać na 11 bitach, czyli ma wielkość 11.

Zdefiniujmy teraz funkcje $f_k : \mathbb{N} \to \mathbb{N}$, gdzie k przebiega zbiór liczb naturalnych. Przyjmujemy $f_1(n) = n + 2$ oraz

$$f_k(n) = \underbrace{f_{k-1}(f_{k-1}(f_{k-1}(\ldots}_{n-1}(2))))$$

dla $k \geqslant 2$. Czyli

$$f_2(n) = 2n, \qquad f_3(n) = 2^n, \qquad f_4(n) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}}_{n} \qquad \text{itd.}$$

Można sobie tylko próbować wyobrazić, co dzieje się dalej. Funkcje, dla których istnieje oszacowanie z góry przez pewną funkcję $C \cdot f_k$ (dla $C, k \in \mathbb{N}$), nazywamy funkcjami pierwotnie rekurencyjnymi. Zatem funkcje, które nie są pierwotnie rekurencyjne, to takie, które rosną jeszcze szybciej niż funkcje f_k. Przykład funkcji, która nie jest pierwotnie rekurencyjna, to funkcja $F(n) = f_n(n)$. Czytelnik Wnikliwy może się przekonać, że istotnie dla każdej funkcji $C \cdot f_k$ istnieje takie n, że $F(n) > C \cdot f_k(n)$. Podobnie definiuje się inny znany przykład: funkcję Ackermanna.

Powiemy też, że problem nie jest pierwotnie rekurencyjny, jeśli nie istnieje żaden algorytm dla tego problemu, który dla każdych danych wielkości n wykonuje co najwyżej $C \cdot f_k(n)$ operacji dla pewnych $C, k \in \mathbb{N}$. O dziwo, znamy jednak techniki, które pozwalają nam projektować algorytmy działające w czasie nieograniczonym przez funkcję pierwotnie rekurencyjną. Najpierw przedstawmy jednak głównego bohatera, czyli badany problem.

Będziemy rozważać wektory składające się z n liczb naturalnych, czyli elementy zbioru $\mathbb{N}^n$. Dany jest wektor początkowy $v_p \in \mathbb{N}^n$, wektor końcowy $v_k \in \mathbb{N}^n$ oraz skończony zbiór wektorów krokowych $K \subseteq \mathbb{Z}^n$. Zwróćmy uwagę na to, że w wektorach krokowych dopuszczamy ujemne współrzędne! Pytanie brzmi: czy możemy w skończonej liczbie ruchów dojść z punktu v_p do punktu v_k, o ile mamy do dyspozycji następujące ruchy:

1. dodanie wektora krokowego, o ile to nie spowoduje, że wyjdziemy poza $\mathbb{N}^n$: $v \mapsto v + v'$, gdzie $v' \in K$, $v + v' \in \mathbb{N}^n$;
2. zmniejszenie którejś współrzędnej, o ile to nie spowoduje, że wyjdziemy poza $\mathbb{N}^n$: $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_{i-1}, x_i - c, x_{i+1}, \ldots, x_n)$, gdzie $x_i \geqslant c$, $c \in \mathbb{N}$.

Jeśli istnieje takie przejście, to nazwiemy je *ścieżką* z v_p do v_k.

Rozwiązanie zadania M 1410.
Mamy $a_2 = 1^{\pm 1} = 1$, $a_3 = 1^{\pm 1} + 1^{\pm 1} = 2$. Oczywiście, dla każdego n zachodzi $a_n \geqslant 1$, a więc również $a_n^{-1} \leqslant a_n$. W takim razie dla $n \geqslant 4$ mamy

$$1 + 1 + \frac{1}{2} + \frac{1}{a_4} + \ldots + \frac{1}{a_{n-1}} \leqslant$$
$$\leqslant a_n \leqslant 1 + 1 + 2 + a_4 + \ldots + a_{n-1}.$$

Z drugiej nierówności możemy przez łatwy dowód indukcyjny wyprowadzić własność $a_n \leqslant 2^{n-2}$ dla $n \geqslant 2$. Wówczas jednak pierwsza nierówność daje oszacowanie

$$a_{2014} \geqslant 2 + \frac{1}{2} + \frac{1}{2^2} + \ldots + \frac{1}{2^{2014-3}} =$$
$$= 3 - \frac{1}{2^{2011}}.$$

Wartość $3 - 2^{-2011}$ jest osiągana przez a_{2014} dla ciągu określonego przez warunek $a_k = 2^{k-2}$ dla $k \leqslant 2013$. Faktycznie wówczas

$$a_n = a_1 + \ldots + a_{n-1}$$

dla $n \leqslant 2013$, oraz

$$a_{2014} = a_1^{-1} + \ldots + a_{2013}^{-1},$$

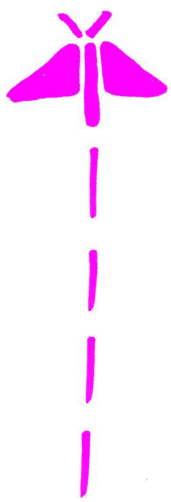
więc ciąg ten spełnia warunki zadania.

[*] Uniwersytet w Bayreuth;
Instytut Informatyki,
Uniwersytet Warszawski

Dla powyższego problemu można podać algorytm działający w czasie rzędu 2^{2^n} (czyli ten problem jest pierwotnie rekurencyjny). Ale właśnie ten przykład świetnie ilustruje techniki, za pomocą których rozwiązuje się nawet dużo trudniejsze problemy, w tym takie, które nie są pierwotnie rekurencyjne.

Na początek przedstawimy nasze główne narzędzie: lemat Dicksona. Powiemy, że wektor $v = (x_1, \ldots, x_n)$ jest *mniejszy lub równy* niż wektor $v' = (x'_1, \ldots, x'_n)$, jeśli $x_i \leqslant x'_i$ dla każdej współrzędnej $i \in \{1, \ldots, n\}$; piszemy wtedy $v \preceq v'$.

Lemat 1 (Dicksona)**.** *Niech $v^1, v^2, \ldots$ będzie nieskończonym ciągiem wektorów z $\mathbb{N}^n$. Wówczas istnieją pewne dwa indeksy $i < j$, takie, że $v^i \preceq v^j$.*



Algorytm sprawdzający, czy istnieje ścieżka z v_p do v_k, składa się z dwóch procedur działających równocześnie (lub – jak kto woli – wykonujących kroki na zmianę). Pierwsza – nazwijmy ją pozytywną – usiłuje znaleźć ścieżkę z v_p do v_k (czyli konstruktywny dowód na to, że taka istnieje), druga natomiast – nazwijmy ją negatywną – próbuje znaleźć dowód, że nie ma żadnej ścieżki z v_p do v_k. Istotne jest, żebyśmy dobrze zaprojektowali obie procedury: jeśli szukana ścieżka istnieje, to procedura pozytywna musi w pewnym momencie ją znaleźć, a jeśli ścieżka nie istnieje, to procedura negatywna ma w pewnym momencie otrzymać dowód tego faktu. Wówczas w każdym przypadku jedna z procedur się zatrzyma i nasz algorytm zakończy się, zwracając właściwą odpowiedź. Zauważmy, że *a priori* nie mamy żadnego oszacowania na to, kiedy program zakończy działanie, i dla niektórych problemów rzeczywiście oczekiwanie na ten moment będzie trwać bardzo długo.

Zaprojektowanie procedury pozytywnej jest łatwe. Po prostu sprawdzane są wszystkie możliwe ciągi ruchów, zaczynając od najkrótszych, a potem przegląda się coraz dłuższe. Dla każdego z nich testujemy, czy może przypadkiem prowadzi on z v_p do v_k i nie powoduje nigdzie po drodze zejścia poniżej zera.

Problem leży w konstrukcji procedury negatywnej. Jak w ogóle może wyglądać dowód nieistnienia ścieżki między danymi punktami? Rozważmy zbiór V wszystkich takich wektorów v, że istnieje z nich ścieżka do wektora końcowego v_k. Zauważmy, że zbiór V jest *zamknięty w górę*: jeśli $v \in V$ oraz $v \preceq v'$ dla pewnego wektora v', to również $v' \in V$. Istotnie, możemy bowiem z v' łatwo dojść do v (zmniejszając niektóre współrzędne), a stąd do v_k. Jeżeli z v_p nie da się dojść do v_k, to zbiór V spełnia następujące warunki:

- $v_p \notin V$, $v_k \in V$;
- V jest zamknięty w górę;
- nie da się wejść spoza V do V.

Taki zbiór nazwiemy *separatorem*. Dodatkowo, zauważmy, że jeśli istnieje jakikolwiek separator, to nie da się dojść z v_p do v_k. Tu właściwie wystarczą tylko pierwszy i trzeci warunek, drugi przyda się jedynie do szukania separatora. Podsumowując, separator istnieje wtedy i tylko wtedy, gdy nie ma ścieżki z v_p do v_k. Procedura negatywna będzie zatem szukała separatora.

Jak to zrealizować? Nie można przecież przeszukiwać wszystkich zbiorów wektorów. Po pierwsze, taki zbiór może być nieskończony (czyli nie damy rady zapamiętać wszystkich elementów). Poza tym nie da się wszystkich zbiorów ustawić w ciąg i posprawdzać po kolei (jest ich nieprzeliczalnie wiele) i właściwie nie wiadomo, jak testować warunek drugi i trzeci. Musimy koniecznie znaleźć przynajmniej jakiś sposób zapamiętania takiego zbioru w skończonej pamięci. Teraz właśnie przychodzi nam z pomocą lemat Dicksona.

Rozważmy zbiór minimalnych wektorów danego separatora – wektor minimalny to taki, że separator nie zawiera elementów mniejszych od niego w porządku $\preceq$. Dzięki lematowi Dicksona wiemy, że jest ich skończenie wiele. Przypuśćmy bowiem, że mamy nieskończenie wiele wektorów minimalnych. Ustawiamy je w ciąg $v^1, v^2, \ldots$ i z lematu Dicksona dostajemy, że $v^i \preceq v^j$ dla pewnych indeksów $i < j$. Mamy sprzeczność z minimalnością v^j! Zatem każdy separator składa się ze skończenie wielu elementów minimalnych $v^1, v^2, \ldots, v^j$ oraz wszystkich wektorów większych od któregoś z nich. Możemy go zatem reprezentować poprzez zbiór $\{v^1, \ldots, v^j\}$.

Rozwiązanie zadania F 848.
Nachylenie drogi powoduje zmniejszenie siły nacisku N opon na podłoże do wartości $N = mg\cos\alpha$, gdzie m to masa samochodu, a g przyspieszenie ziemskie. Maksymalna wartość siły tarcia T wynosi więc $T = fmg\cos\alpha$. Jednocześnie pojawia się siła zsuwająca $F = mg\sin\alpha$ przeciwdziałająca sile tarcia. Hamowanie (zmniejszenie prędkości) możliwe jest tylko, gdy $f\cos\alpha > \sin\alpha$. Podczas hamowania od prędkości v do zatrzymania samochód porusza się z opóźnieniem $a = fg\cos\alpha - g\sin\alpha$ i przebywa drogę

$$s(\alpha) = \frac{v_2}{2g(f\cos\alpha - \sin\alpha)},$$

a więc $s(\alpha) > 2s(0)$, gdy

$$f(2\cos\alpha - 1) < 2\sin\alpha.$$

Po skorzystaniu ze znanych tożsamości trygonometrycznych nierówność tę można sprowadzić do postaci

$$3\operatorname{tg}^2\frac{\alpha}{2} + \frac{4}{f}\operatorname{tg}\frac{\alpha}{2} - 1 > 0.$$

Dla suchej nawierzchni $f \approx 1$ i $f \approx 0{,}3$ na nawierzchni mokrej (guma po betonie). Prowadzi to do warunków: $\alpha > 24{,}3^\circ$ na suchej nawierzchni i $\alpha > 8{,}4^\circ$ na nawierzchni mokrej.

Procedura negatywna przeszukuje więc wszystkie skończone zbiory wektorów $\{v^1, \ldots, v^j\}$ i sprawdza, czy zbiór wszystkich wektorów większych od któregoś z nich nie jest separatorem. Z podanych powyżej trzech warunków drugi jest spełniony zawsze. Pierwszy sprawdza się łatwo, wystarczy porównać współrzędne wektorów v_p, v_k i wektorów reprezentujących separator. Trzeci również można rozstrzygać w miarę nietrudno – Czytelnik Ambitny może śmiało spróbować to udowodnić.

Tym samym zakończyliśmy opis obu procedur, a więc i algorytmu. Najistotniejszym spostrzeżeniem było, że elementów minimalnych w separatorze jest skończenie wiele. To typowa sytuacja: kluczem do zaprojektowania algorytmu, który zawsze się zatrzymuje, ale, być może, działa bardzo wolno, jest obserwacja, że pewien obiekt pojawiający się w analizie problemu jest skończony. Do otrzymywania takich wyników nadaje się znakomicie lemat Dicksona lub następujący lemat Higmana, stosowany w przypadku słów, a nie wektorów.

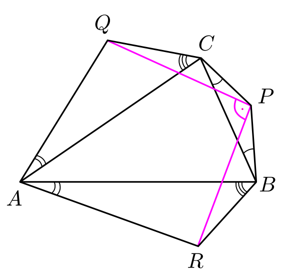
Lemat 2 (Higmana). *Niech $w^1, w^2, \ldots$ będzie nieskończonym ciągiem słów nad skończonym alfabetem. Wówczas istnieją takie indeksy $i < j$, że słowo w^i zawiera się w słowie w^j (czyli ciąg liter słowa w^i jest podciągiem ciągu liter słowa w^j).*

Na pierwszy rzut oka widać podobieństwo do lematu Dicksona. Tak naprawdę oba te lematy są szczególnymi przypadkami pewnych bardziej ogólnych twierdzeń. To już jednak temat na zupełnie inną opowieść.

Zadania

Redaguje Tomasz TKOCZ

M 1408. Udowodnić, że dowolna liczba rzeczywista x spełnia nierówność
$$|x+1| + |x+2| + \ldots + |x+2014| \geqslant 1007^2.$$
Rozwiązanie na str. 5

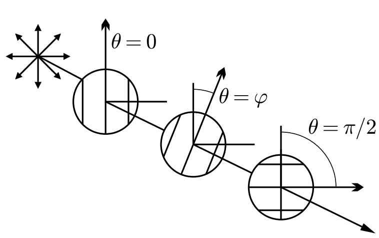


Rys. 1

M 1409. Na zewnątrz trójkąta ABC dane są punkty P, Q, R (rys. 1) wyznaczone przez warunki
$$\sphericalangle BCP = \sphericalangle CBP = 15^\circ, \qquad \sphericalangle BAR = \sphericalangle CAQ = 30^\circ, \qquad \sphericalangle ABR = \sphericalangle ACQ = 45^\circ.$$
Udowodnić, że odcinki PQ i PR są równe i prostopadłe.
Rozwiązanie na str. 10

M 1410. Rozpatrujemy takie ciągi $(a_n)_{n=1}^{\infty}$, że $a_1 = 1$ oraz, dla każdego $n \geqslant 2$,
$$a_n = a_1^{\pm 1} + \ldots + a_{n-1}^{\pm 1}$$
dla pewnego wyboru znaków $+$, $-$ w wykładnikach. Znaleźć najmniejszą możliwą wartość a_{2014}.
Rozwiązanie na str. 6

Przygotowali Michał NAWROCKI i Andrzej MAJHOFER



Rys. 2

F 847. Jak wiadomo, niespolaryzowane światło nie przechodzi przez układ dwóch skrzyżowanych pod kątem prostym polaryzatorów liniowych. Jeżeli jednak pomiędzy te polaryzatory wstawimy pod odpowiednim kątem trzeci polaryzator liniowy (rys. 2), to światło nie będzie wygaszane. Przyjmijmy, że oś polaryzacji wstawionego polaryzatora tworzy kąt φ z osią pierwszego polaryzatora. Jak zależy natężenie światła przechodzącego przez taki układ polaryzatorów od kata φ? Dla jakiego kąta φ będzie ono największe, a dla jakiego najmniejsze?
Rozwiązanie na str. 11

F 848. Przy jakiej wartości α kąta nachylenia drogi do poziomu długość drogi hamowania zjeżdżającego samochodu rośnie ponad dwukrotnie w porównaniu z hamowaniem na odcinku poziomym? Przyjmujemy, że współczynnik tarcia opon o asfalt wynosi f.
Rozwiązanie na str. 7