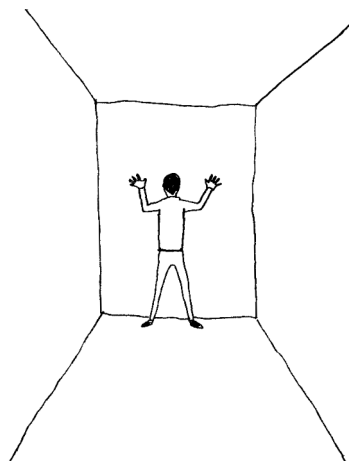


Problemy 3SUM-trudne w geometrii

Jakub RADOSZEWSKI



Zad. 1. Jak rozwiązać problem 3SUM w czasie $O(n^2)$? (Albo chociaż w czasie $O(n^2 \log n)$?)

Zad. 2. W problemie 3SUM elementy a, b, c mogą się powtarzać. Nie ma to jednak większego znaczenia, gdyż w czasie $O(n \log n)$ można łatwo identyfikować instancje problemu, dla których istnieje rozwiązanie z powtarzającymi się elementami. Jak to zrobić?

Algorytm ten jest opisany w pracy: I. Baran, E.D. Demaine, M. Patrascu, *Subquadratic algorithms for 3SUM*, WADS 2005.

Zad. 3. Udowodnij, że $3SUM =_n 3SUM'$, tzn. pokaż, jak wykonać obie redukcje. Przykłady różnych redukcji znajdziesz w dalszej części artykułu.

Wiele z opisanych dalej problemów pochodzi z przystępnej i łatwo dostępnej pracy przeglądowej: J. King, *A survey of 3SUM-hard problems*, 2004.

Kiedy rozwiązujemy jakiś problem informatyczny, często naszym celem jest podanie jak najefektywniejszego algorytmu. Jednak czasem możemy natknąć się przy tym na „ścianę” – danego problemu nie da się rozwiązać tak efektywnie, jak byśmy tego chcieli. Najpowszechniej znanym przykładem opisanego zjawiska jest klasa problemów NP-zupełnych. O problemach z tej klasy (a należy do niej wiele naturalnych i praktycznych zagadnień) podejrzewa się, że nie da się ich rozwiązać w czasie wielomianowym względem rozmiaru danych wejściowych. Niestety, tylko *podejrzewa się*, a rozstrzygnięcie tej hipotezy (znanej też jako „ $P \neq NP$?”) jest obecnie najsłynniejszym otwartym problemem informatyki teoretycznej.

Na szczęście wiele ważnych problemów obliczeniowych umiemy rozwiązywać w czasie wielomianowym. Jednak w praktyce wielomian wielomianowi nierówny – jest przecież istotne, czy dany problem umiemy rozwiązać w czasie $\Theta(n)$, czy też w czasie $\Theta(n^{10})$. Okazuje się, że także w przypadku problemów rozwiązywalnych w czasie wielomianowym (tzw. klasa problemów P) znane są pewne narzędzia pozwalające stwierdzić, że danego problemu *zapeune* nie da się rozwiązać w czasie szybszym niż taki a taki, a jeśli nawet się da, to jest to bardzo trudne.

W tym artykule skupimy naszą uwagę na następującym problemie 3SUM:

Dany jest zbiór S złożony z n liczb całkowitych. Czy istnieją takie liczby $a, b, c \in S$, że $a + b + c = 0$?

Wszystkie omawiane w tym artykule problemy mają charakter decyzyjny, tzn. odpowiedzią w każdym z nich jest pojedyncza wartość logiczna. Na potrzeby naszego artykułu przyjmujemy następującą definicję *redukcji* problemu: Powiemy, że problem A można zredukować do problemu B w czasie $O(f(n))$, jeśli dowolną instancję (czyli dane wejściowe) problemu A o rozmiarze n można przekształcić w czasie $O(f(n))$ w stałą liczbę instancji problemu B o rozmiarze $O(n)$ tak, żeby na podstawie odpowiedzi dla wyznaczonych instancji problemu B można było udzielić odpowiedzi dla rozważanej instancji problemu A . Będziemy wówczas pisać, że $A \preceq_{f(n)} B$, co oznacza tyle, że problem B jest co najmniej tak trudny jak problem A , gdyż jeśli umiemy w czasie $O(f(n))$ rozwiązać problem B , to umiemy w takim samym czasie rozwiązać problem A . Jeśli $A \preceq_{f(n)} B$ oraz $B \preceq_{f(n)} A$, to będziemy pisać, że $A =_{f(n)} B$.

Problemem, który będziemy redukować do innych zagadnień, jest wspomniany już problem 3SUM. Przez wiele lat znany był algorytm rozwiązujący ten problem w czasie $\Theta(n^2)$ (patrz zadanie 1 na marginesie) i uważano, że nie da się go rozwiązać w złożoności mniejszej niż $\Theta(n^2)$. Wprowadzono też klasę problemów 3SUM-trudnych, czyli problemów, do których można zredukować problem 3SUM w czasie $o(n^2)$. Od 2005 roku wiadomo jednak, że problem 3SUM można rozwiązać *trochę* szybciej, a mianowicie w czasie

$$O\left(\frac{n^2(\log \log n)^2}{\log^2 n}\right).$$

Mimo tego usprawnienia wciąż nie widać nadziei na znalezienie algorytmu rozwiązującego ten problem w czasie *istotnie* lepszym niż $\Theta(n^2)$.

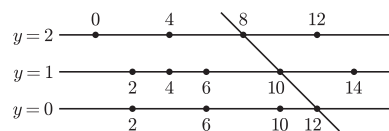
Będziemy także używać następującego bliźniaczego problemu 3SUM', który jest równoważny z problemem 3SUM (tj. $3SUM =_n 3SUM'$):

Dane są trzy zbiory liczb całkowitych A, B, C , zawierające łącznie n elementów. Czy istnieją liczby $a \in A, b \in B$ i $c \in C$, takie że $a + b = c$?

Problemy, które będziemy rozważać w dalszej części tekstu, mają charakter geometryczny. Jako pierwszy rozpatrzmy problem nazywany GEOMBASE – podstawowy problem 3SUM-trudny w geometrii, służący jako narzędzie

do dowodzenia 3SUM-trudności innych problemów. Jest on określony następująco:

Punkt kratowy to punkt o współrzędnych całkowitych.



Rys. 1. Instancja problemu GEOMBASE otrzymana z instancji $A = \{1, 3, 5, 6\}$, $B = \{0, 2, 4, 6\}$, $C = \{2, 4, 6, 10, 14\}$ problemu 3SUM'. Wynikiem jest $a = 6$, $b = 4$, $c = 10$.

Danych jest n punktów kratowych położonych na prostych $y = 0$, $y = 1$ i $y = 2$. Czy istnieje prosta inna niż pozioma przechodząca przez jakies trzy spośród tych punktów?

Po chwili namysłu nietrudno dostrzec tu podobieństwo do problemu 3SUM'. Nie jest to przypadek; zachodzi bowiem $3SUM' =_n GEOMBASE$. Przeprowadzimy tylko redukcję $3SUM' \preceq_n GEOMBASE$, redukcja w drugą stronę jest właściwie taka sama. Wystarczy mianowicie dla każdej liczby $a \in A$ wybrać punkt $(2a, 0)$ na prostej $y = 0$, dla każdej liczby $b \in B$ – punkt $(2b, 2)$, a dla każdej liczby $c \in C$ – punkt $(c, 1)$, patrz rysunek 1. Punkty $(2a, 0)$, $(c, 1)$ i $(2b, 2)$ są współliniowe, gdy

$$c - 2a = 2b - c,$$

czyli dokładnie gdy $c = a + b$. Stąd odpowiedź dla otrzymanej instancji problemu GEOMBASE jest pozytywna wtedy i tylko wtedy, gdy pozytywna jest odpowiedź dla odpowiadającej jej instancji problemu 3SUM', a to jest dokładnie to, co chcieliśmy uzyskać.

Problem GEOMBASE, jakkolwiek użyteczny, nie jest sam w sobie nadmiernie interesujący. Dużo ciekawszy jest podobny do niego następujący problem 3POINTSONLINE:

Na płaszczyźnie danych jest n punktów. Czy istnieje prosta przechodząca przez jakies trzy spośród tych punktów?

Ciekawe, ilu spośród Czytelników próbowało kiedykolwiek rozwiązać ten problem w czasie zbliżonym do $\Theta(n \log n)$ (sam autor wielokrotnie próbował).

Ten problem także jest 3SUM-trudny; tym razem redukcję przeprowadzimy wprost z problemu 3SUM. Niech S będzie instancją problemu 3SUM. Dla każdej liczby $x \in S$ rysujemy na płaszczyźnie punkt (x, x^3) . Zbadajmy, kiedy trzy różne tak otrzymane punkty (a, a^3) , (b, b^3) i (c, c^3) są współliniowe:

$$\begin{aligned} \frac{b^3 - a^3}{b - a} = \frac{c^3 - b^3}{c - b} &\Leftrightarrow \frac{(b - a)(b^2 + ab + a^2)}{b - a} = \frac{(c - b)(c^2 + bc + b^2)}{c - b} \Leftrightarrow \\ &\Leftrightarrow b^2 + ab + a^2 = c^2 + bc + b^2 \Leftrightarrow \\ &\Leftrightarrow a^2 - c^2 + ab - bc = 0 \Leftrightarrow \\ &\Leftrightarrow (a - c)(a + c) + b(a - c) = 0 \Leftrightarrow \\ &\Leftrightarrow (a - c)(a + b + c) = 0 \Leftrightarrow \\ &\Leftrightarrow a + b + c = 0. \end{aligned}$$

Okazuje się zatem, że punkty (a, a^3) , (b, b^3) i (c, c^3) są współliniowe wtedy i tylko wtedy, gdy liczby a, b, c stanowią rozwiązanie problemu 3SUM.

W ten sposób otrzymujemy żadaną redukcję problemu 3SUM do problemu 3POINTSONLINE, pod warunkiem, że w tym pierwszym problemie interesują nas tylko trójki różnych liczb. To jednak możemy zapewnić (patrz zad. 2).

Wytrawny znawca geometrii natychmiast zauważy, że 3SUM-trudność problemu 3POINTSONLINE implikuje bezpośrednio 3SUM-trudność dualnego problemu POINTON3LINES:

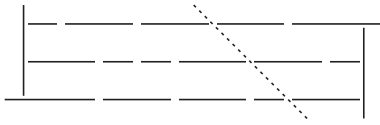
Na płaszczyźnie danych jest n prostych. Czy któreś trzy proste przecinają się w jednym punkcie?

Wróćmy teraz do problemu GEOMBASE. Skoro reklamowaliśmy go jako przydatne narzędzie w geometrii, to najwyższy czas, aby podać kilka jego zastosowań. Wykazaliśmy już, że jest to problem 3SUM-trudny. Jeśli zatem pokażemy redukcję z tego problemu do jakiegoś innego problemu geometrycznego działającą w czasie $o(n^2)$, to w ten sposób wykażemy, że ów drugi problem również jest 3SUM-trudny. Pierwszym problemem z tej serii, który weźmiemy pod lupę, jest problem o nazwie SEPARATOR:

Na płaszczyźnie danych jest n odcinków. Czy istnieje prosta nieprzecinająca żadnego z nich, dzieląca zbiór tych odcinków na dwa niepuste podzbiory? Jeśli chcemy, możemy założyć, że podane odcinki są parami rozłączne.

Głównym zastosowaniem problemu 3POINTSONLINE wydaje się weryfikowanie częstego w problemach geometrycznych założenia, że żadne trzy spośród zadanych punktów nie leżą na jednej prostej.

Zad. 4. Określmy przekształcenie geometryczne T , które punktowi $p = (a, b)$ przyporządkowuje prostą $T(p) : y = ax - b$, a prostej $l : y = kx + d$ przyporządkowuje punkt $T(l) = (k, -d)$. Udowodnij, że punkt p leży na prostej l wtedy i tylko wtedy, gdy prosta $T(p)$ przechodzi przez punkt $T(l)$. (Więcej na temat tej dualności Czytelnik znajdzie w numerze 8/2013 *Delty*.)



Rys. 2. Redukcja instancji problemu GEOMBASE z rysunku 1 do instancji problemu SEPARATOR.

Zad. 5. Jak dobrać ε w opisanej redukcji?

Sprowadzimy problem GEOMBASE do problemu SEPARATOR. Tym razem redukcja będzie działać w czasie $O(n \log n)$, gdyż wymagać będzie sortowania. Posortujemy punkty na każdej z prostych poziomych i połączymy pary kolejnych punktów odcinkami w taki sposób, żeby między każdymi dwoma kolejnymi odcinkami występował bardzo mały odstęp, o długości równej $\varepsilon > 0$. Widzimy już, że przy tej konstrukcji prosta rozdzielająca zbiór odcinków, przechodząca przez trzy odstępy, odpowiada trójce punktów współliniowych należących do różnych prostych poziomych. Należy jeszcze zadbać o wyeliminowanie poziomych prostych rozdzielających. Wystarczy w tym celu umieścić dwa pionowe odcinki po bokach konstrukcji i ustawić je „na zakładkę” z odcinkami poziomymi, tak aby z kolei wyeliminować trywialne proste rozdzielające pionowe (rys. 2). W ten sposób wykazaliśmy, że $\text{GEOMBASE} \preceq_{n \log n} \text{SEPARATOR}$.

Kolejny przykład. Okazuje się, że problem GEOMBASE możemy zredukować do następującego, całkiem naturalnego i mającego liczne zastosowania praktyczne problemu **PLANARMOTIONPLANNING**:

Na płaszczyźnie danych jest n odcinków-przeszkód i jeden odcinek reprezentujący robota. Czy robot może się przedostać z zadanej pozycji początkowej do zadanej pozycji końcowej, przemieszczając się jedynie za pomocą ciągłych przesunięć i obrotów bez dotykania żadnej z przeszkód? Znow, jeśli chcemy, możemy założyć, że odcinki-przeszkody są parami rozłączne.

Redukcja jest tutaj podobna jak poprzednio. Przeszkodami będą odcinki poziome z wcześniejszej redukcji. Robota umieścimy poniżej całej konstrukcji, a jego celem będzie przedostanie się powyżej konstrukcji. Sam robot będzie na tyle długi, żeby nie mógł przedostać się na drugą stronę inaczej niż tylko przez trzy odstępy naraz. Należy jeszcze zadbać, aby robot nie mógł ominąć całej konstrukcji bokiem. W tym celu trzeba poniżej i powyżej konstrukcji zbudować „klatki” uniemożliwiające ucieczkę robota do „nieskończoności”. W ten sposób otrzymujemy wniosek, że $\text{GEOMBASE} \preceq_{n \log n} \text{PLANARMOTIONPLANNING}$.

Zad. 6. Jak zbudować klatki w opisanej redukcji? (Narysuj!)

To, co przedstawiliśmy do tej pory, to tylko niewielka grupa znanych problemów 3SUM-trudnych. Dla Niestrudzonych Czytelników mamy w zanadrzu jeszcze kilka innych takich problemów – tym razem już dowody pominiemy. Zaczniemy od jeszcze jednego pomocniczego problemu 3SUM-trudnego **SEGMENTSCONTAINPOINTS**:

Na prostej danych jest n punktów i $m = O(n)$ parami rozłącznych przedziałów. Czy istnieje wektor, o który można przesunąć wszystkie zadane punkty, tak aby każdy z nich znalazł się w jakimś przedziale?

Problem ten można zredukować do każdego z następujących zagadnień typu **POLYGONCONTAINMENT**:

Na płaszczyźnie dane są dwa wielokąty, odpowiednio o n i $m = O(n)$ bokach. Mamy do dyspozycji pewien zestaw izometrii płaszczyzny. Czy za pomocą tych izometrii możemy pierwszy z tych wielokątów umieścić wewnątrz drugiego, przy założeniu, że:

- (a) wielokąty są dowolne i dopuszczamy dowolne przesunięcia;
- (b) wielokąty są wypukłe i dopuszczamy dowolne obroty;
- (c) wielokąty są wypukłe i dopuszczamy dowolne obroty i przesunięcia?

A już na sam koniec kilka problemów 3SUM-trudnych typu **COVERING**:

- (a) Czy dany zbiór n dwustronnie nieskończonych pasków (o różnych szerokościach i kierunkach) pokrywa zadany prostokąt?
- (b) Czy dany zbiór n trójkątów pokrywa zadany trójkąt?
- (c) Czy pole sumy danych n trójkątów jest mniejsze od ustalonej liczby?
- (d) Czy suma danych n trójkątów zawiera jakąś dziurę?
- (e) Danych jest n półpłaszczyzn. Czy istnieje punkt płaszczyzny pokryty przez co najmniej k spośród tych półpłaszczyzn?

Redukcje dotyczące problemów **SEGMENTSCONTAINPOINTS** i **POLYGONCONTAINMENT** są opisane w pracy: G. Barequet, S. Har-Peled, *Polygon-containment and translational min-Hausdorff-distance between segment sets are 3SUM-hard*, SODA 1999.

Zad. 7. Opisz redukcję problemu **SEGMENTSCONTAINPOINTS** do problemu **POLYGONCONTAINMENT** w wersji (a). Redukcja powinna działać w czasie $O(n \log n)$. Wskazówka: szukane wielokąty mają kształt grzebieni.

Zad. 8. Opisz redukcję wariantu (a) problemu **COVERING** do wariantu (b).

Redukcje do problemów typu **COVERING** można znaleźć w podstawowej (tj. pionierskiej) pracy dotyczącej 3SUM-trudności: A. Gajentaan, M.H. Overmars, *On a class of $O(n^2)$ problems in computational geometry*, Comp. Geom. 5(3), 1995.