Informatyczny kącik olimpijski (??): Czworokąty wypukłe

W tym kąciku zajmiemy się zadaniem *Quadrilaterals* z obozu w Petrozawodzku w 2006 roku. Na płaszczyźnie dane jest n punktów w położeniu ogólnym (tzn. żadna trójka punktów nie leży na jednej prostej). Należy wyznaczyć liczbę czworokątów wypukłych, których wierzchołki znajdują się wśród podanych punktów.

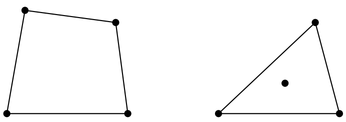
Aby sprawdzić, czy punkty p_0, p_1, p_2, p_3 mogą być kolejnymi wierzchołkami czworokąta wypukłego w porządku przeciwnym do kierunku ruchu wskazówek zegara, wystarczy zbadać, czy każda trójka kolejnych punktów tworzy zakręt w lewo. W tym celu można zbadać znak iloczynu wektorowego:

$$(p_i - p_{(i-1) \bmod 4}) \times (p_{(i+1) \bmod 4} - p_i) > 0.$$

Formalnie iloczyn wektorowy jest zdefiniowany w przestrzeni trójwymiarowej; w zadaniach z geometrii w dwóch wymiarach zwykle określa się go dla wektorów $v_1 = [x_1, y_1]$, $v_2 = [x_2, y_2]$ jako liczbę $x_1 y_2 - x_2 y_1$.

Sprawdzając wszystkie możliwe czwórki uporządkowane, liczbę „dobrych czwórek” możemy wyznaczyć w czasie $O(n^4)$. Liczbę czworokątów wypukłych uzyskamy, dzieląc liczbę „dobrych czwórek” przez 4 (bierzemy poprawkę na obroty cykliczne).

Aby otrzymać szybsze rozwiązanie, skorzystamy z pewnego pomysłowego triku. Rozważmy dowolną nieuporządkowaną czwórkę punktów (w położeniu ogólnym) i zastanówmy się, jakie czworokąty można na niej zbudować. W zależności od wzajemnego ustawienia punktów mamy dwa przypadki (patrz rys. 1): albo wszystkie punkty leżą na otoczce wypukłej (przypadek A), albo trzy punkty leżą na otoczce wypukłej, a czwarty punkt w środku (przypadek B). Łatwo przekonać się, że w pierwszym przypadku czwórka punktów wyznacza jeden czworokąt wypukły, natomiast w drugim przypadku dostajemy trzy różne czworokąty wklęsłe.



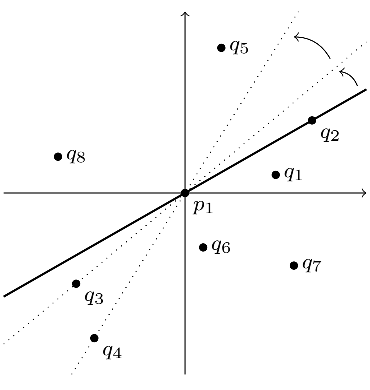
Rys. 1. Dwa przypadki położenia nieuporządkowanej czwórki punktów: A (po lewej) i B (po prawej).

Powiemy, że para punktów $\{p_1, p_2\}$ tworzy *przecięcie* z parą punktów $\{q_1, q_2\}$, jeśli punkty q_1, q_2 leżą po różnych stronach prostej wyznaczonej przez punkty p_1, p_2. Zauważmy, że czwórka punktów z przypadku A generuje nam dwa przecięcia (gdy za punkty p_1, p_2 weźmiemy przeciwległe wierzchołki czworokąta wypukłego), natomiast w przypadku B są to trzy przecięcia (gdy jeden z punktów p_1, p_2 leży w środku otoczki wypukłej). Niech P będzie liczbą wszystkich przecięć w zadanym zbiorze n punktów, natomiast a i b będą liczbą czwórek punktów tworzących układy odpowiednio typu A i typu B. Możemy napisać następujący układ równań:

$$\begin{cases} \binom{n}{4} = a + b \\ P = 2a + 3b \end{cases}$$

Rozwiązując ten układ, dostajemy, że liczba czworokątów wypukłych to $a = 3\binom{n}{4} - P$ (przy okazji: liczba wszystkich czworokątów to $a + 3b = 2P - 3\binom{n}{4}$).

Pozostaje pokazać, jak wyznaczyć liczbę P. Jeśli dla ustalonej pary punktów $\{p_1, p_2\}$ dokładnie x punktów leży po jednej stronie prostej p_1p_2 i y punktów leży po drugiej stronie tej prostej, to mamy $x \cdot y$ przecięć zawierających parę $\{p_1, p_2\}$. Rozważając każdą taką parę $\{p_1, p_2\}$ osobno, uzyskujemy czas $O(n^3)$.



Rys. 2. Przejście prostej ze stanu $(i, x, y) = (2, 2, 5)$ do stanu $(3, 2, 5)$, a następnie do $(4, 3, 4)$.

Szybciej wyznaczymy P, korzystając z metody zamiatania. Ustalmy punkt p_1 i przesuńmy punkty na płaszczyźnie tak, żeby środek układu współrzędnych znalazł się w p_1. Dla każdego innego punktu q wyznaczmy kąt $\alpha(q) \in [0, 2\pi)$, jaki tworzy *półprosta* p_1q z osią OX. Niech $q_1, \ldots, q_{n-1}$ będzie listą tych punktów posortowaną rosnąco po kątach $\alpha(q) \bmod \pi$, czyli po kątach, jakie tworzy *prosta* p_1q z osią OX. Na początku przyjmujemy $p_2 = q_1$ i wyznaczamy liczbę x i y punktów, które leżą powyżej i poniżej prostej p_1p_2 (tj. odpowiednio na lewo i na prawo od wektora $\overrightarrow{p_1p_2}$). W każdym kolejnym kroku $i = 1, \ldots, n-2$ zmieniamy punkt p_2 na q_{i+1} i uaktualniamy x i y (patrz rys. 2):

if $\alpha(q_i) < \pi$ **then** $y := y + 1$ **else** $x := x + 1$;
if $\alpha(q_{i+1}) < \pi$ **then** $x := x - 1$ **else** $y := y - 1$;

Ponieważ dla ustalonego punktu p_1 sortowanie zajmie czas $O(n \log n)$, a przejrzenie – czas $O(n)$, więc cały algorytm wykona się w czasie $O(n^2 \log n)$. Uzyskany wynik (sumę iloczynów $x \cdot y$) musimy podzielić przez 2, gdyż interesują nas nieuporządkowane pary punktów $\{p_1, p_2\}$.

Tomasz IDZIASZEK