Informatyczny kącik olimpijski (48): Zliczanie prostokątów

W tym kąciku rozwiążemy zadanie *Prostokąty* (*Rectangles*), które pojawiło się w 2007 roku na konkursie organizowanym przez MIT w serwisie `spoj.pl`.

Zadanie brzmi tak: na płaszczyźnie danych jest n różnych punktów. Na ile sposobów można wybrać cztery z nich tak, aby były one wierzchołkami prostokąta o bokach równoległych do osi układu współrzędnych?

Pierwsze rozwiązanie, które powinno przyjść nam do głowy, wykorzystuje obserwację, że każdy szukany prostokąt jest jednoznacznie określony przez jego dwa przeciwległe wierzchołki. Możemy więc dla każdej pary danych punktów (lewy dolny, prawy górny) sprawdzić, czy do zbioru należą dwa pozostałe wierzchołki. Jeśli do sprawdzania przynależności do zbioru wykorzystamy tablicę haszującą, takie rozwiązanie będzie działało w czasie $O(n^2)$.

Może się wydawać, że sporo czasu w powyższym rozwiązaniu tracimy na sprawdzaniu „złych" par – przekątnych, dla których pozostałe dwa punkty nie istnieją. Niestety, prosty przykład n punktów ustawionych w „dwuszereg" pokazuje, że wynik może być rzędu $\Theta(n^2)$, zatem żaden algorytm zliczający prostokąty pojedynczo nie może być istotnie lepszy od powyższego.

Jak zatem znaleźć lepsze rozwiązanie? Zauważmy na początek, że możemy przenumerować współrzędne na wejściu tak, aby wszystkie były naturalne i nie większe niż n. Nie interesują nas bowiem długości boków czy pola prostokątów, tylko ich liczba. Podzielmy punkty na pionowe bloki, tj. ze względu na współrzędną x. Bloki natomiast podzielmy na dwie kategorie – *małe*, czyli te, które zawierają co najwyżej $\lfloor\sqrt{n}\rfloor$ punktów, i *duże*, czyli pozostałe.

Zapomnijmy na moment o dużych blokach. Powiemy, że współrzędna y należy do bloku, jeśli w bloku znajduje się punkt o drugiej współrzędnej równej y. Będziemy iterować po możliwych położeniach dolnego boku prostokąta, od góry do dołu. Załóżmy, że dolny bok prostokąta ma drugą współrzędną równą y_1, i rozważmy te bloki, do których należy punkt o takiej współrzędnej – punkty w pozostałych blokach nie mają szans należeć do takiego prostokąta. Ignorujemy także wszystkie punkty o drugich współrzędnych nie większych niż y_1 – te także nie mają wpływu na wynik. Teraz wystarczy policzyć, ile jest poziomych odcinków o obu końcach wśród punktów, które nam pozostały – jako że każdy blok zawiera y_1, każdy taki odcinek odpowiada jednemu prostokątowi. Ten krok można wykonać łatwo, przechodząc po wszystkich punktach, które nam pozostały (w dowolnej kolejności, np. od lewej do prawej), przy okazji zliczając w tablicy (współrzędne są małe!), ile razy dana rzędna już wystąpiła, i aktualizując wynik.

Załóżmy, że wielkości małych bloków to $b_1, b_2, \ldots$, przy czym $b_i \leqslant \lfloor\sqrt{n}\rfloor$ i $\sum b_i \leqslant n$. Przy uważnej implementacji algorytmu małych bloków wykonamy w nim liczbę operacji co najwyżej rzędu $\sum b_i^2$. Faktycznie, wybrana współrzędna y_1 będzie należała do i-tego bloku b_i razy, i wówczas w fazie przeglądania każdy punkt tego bloku odwiedzimy co najwyżej raz. Wykażemy, że $\sum b_i^2 = O(n\sqrt{n})$. Spróbujmy znaleźć „najgorszy możliwy" ciąg b_i. Jeśli dla pewnych różnych k, l mamy $0 < b_k \leqslant b_l < \lfloor\sqrt{n}\rfloor$, to zmniejszając b_k o 1 i zwiększając b_l o 1, nie zmieniamy $\sum b_i$, a jednocześnie zwiększamy $\sum b_i^2$. Jest to konsekwencja wypukłości funkcji kwadratowej. Powtarzając tę operację dostatecznie dużo razy, otrzymamy wszystkie liczby b_i – być może z wyjątkiem jednej – równe $\lfloor\sqrt{n}\rfloor$ lub 0. Tych niezerowych będzie co najwyżej $\sqrt{n}+1$, ponieważ $\sum b_i \leqslant n$. Ostatecznie, dostajemy

$$\sum b_i^2 \leqslant (\sqrt{n}+1)(\sqrt{n})^2 = O(n\sqrt{n}).$$

Pozostało nam zająć się dużymi blokami. Dla dowolnego bloku B, łatwo policzyć w czasie $O(n)$, ile jest prostokątów, które mają dwa punkty w tym bloku – jeśli dla każdego pozostałego bloku B_k znamy liczbę współrzędnych c_k wspólnych dla B i B_k, to szukaną przez nas liczbą jest $\sum \frac{c_k(c_k-1)}{2}$. Ponieważ tę sumę dodamy do wyniku dla każdego dużego bloku, więc prostokąty o wszystkich punktach w dużych blokach policzymy dwukrotnie... To jednak nie jest duży problem – aby się z nim uporać, wystarczy dla każdego dużego bloku zawrzeć w sumie wszystkie małe bloki i tylko te duże, które znajdują się na lewo od niego. Na szczęście, duże bloki są wystarczająco liczne, aby nie mogło ich być więcej niż $\sqrt{n}$ – a więc ta faza działa także w czasie $O(n\sqrt{n})$.

W powyższych rozważaniach zapomnieliśmy o kosztach związanych z sortowaniem i przenumerowaniem wejściowych współrzędnych. Te nie powinny wynieść jednak więcej niż $O(n \log n)$ operacji, więc są znikome w porównaniu z kosztem faz zliczających. Całe rozwiązanie działa więc w czasie $O(n\sqrt{n})$ i pamięci $O(n)$.

Zainteresowanemu Czytelnikowi polecamy zastanowienie się, co by było, gdybyśmy chcieli, zamiast prostokątów, zliczać kwadraty. Czy umiemy to zrobić szybciej, czy potrzebujemy bardziej skomplikowanych struktur danych? Tak sformułowane zadanie pojawiło się na Obozie Naukowo-Treningowym im. A. Kreczmara w 2010 roku pod tytułem *Ogródek* i można je rozwiązywać w serwisie `main.edu.pl`.

Adam KARCZMARZ
student, Wydział Matematyki, Informatyki i Mechaniki
Uniwersytetu Warszawskiego