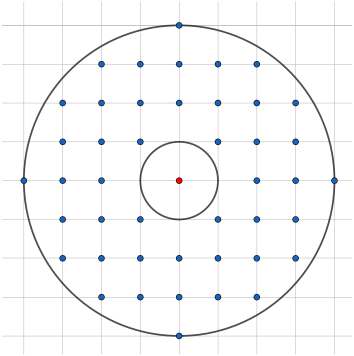
Zadanie L
Obwarzanek

Zgłoszeń: 11
Zaakceptowanych: 2

Pierwsze rozwiązanie:
uw1 [Kondraciuk, Skiba, Paluszek]
02:53:11

Autor: Krzysztof Maziarz

Dla punktu kratowego (a, b) i liczb całkowitych nieujemnych L oraz R, rozważamy zbiór punktów kratowych których odległość od (a, b) jest w przedziale $(L, R]$. Każdy zbiór który można uzyskać w opisany sposób dla pewnych a, b, L, R nazywamy *obwarzankiem*.



Zaczynamy od pustego zbioru punktów S, i n razy dodajemy do niego nowy punkt kratowy. Po każdej takiej operacji należy stwierdzić, czy obecny zbiór S jest obwarzankiem.

Dodatkowo, mamy *bardzo mało dostępnej pamięci* – nie jesteśmy nawet w stanie utrzymywać w pamięci zbioru S.

Za to mamy gwarancję, że wszystkie punkty mają niezbyt duże współrzędne (na moduł nie przekraczające $C = 5000$).

Możemy utrzymywać najmniejszą i największą współrzędną x po punktach znajdujących się w S (l_x, r_x), oraz analogicznie dla y (l_y, r_y).

Warunkiem koniecznym aby obecny zbiór punktów był obwarzankiem jest $r_x - l_x = r_y - l_y$. Ponadto, możemy wyliczyć zewnętrzny promień

$$R = \frac{r_x - l_x}{2}$$

oraz środek

$$(a, b) = \left(\frac{l_x + r_x}{2}, \frac{l_y + r_y}{2} \right)$$

Załóżmy, że dla każdego $r \in \{1, 2, ..., 5000\}$ znamy liczbę punktów kratowych zawartych w kole o środku w $(0, 0)$ i promieniu r. Wtedy mając wyliczony promień zewnętrzny R, i znając liczbę wczytanych punktów, możemy jednoznacznie wyznaczyć L.

Znamy już teraz wartości a, b, L, R – albo S jest dokładnie takim obwarzankiem, albo nie jest obwarzankiem wcale.

By odpowiedzieć na pytanie czy S faktycznie jest obwarzankiem o zadanych parameterach, musimy utrzymywać jeszcze jeden rodzaj informacji. Niech p będzie liczbą pierwszą, a $h(x, y)$ pewną funkcją hashującą $\mathbb{Z}^2 \to \mathbb{Z}_p$.

Zbiór punktów S hashujemy sumując h po jego elementach:

$$hash(S) = \left(\sum_{(x,y)\in S} h(x, y) \right) \mod p$$

Zakładając, że $h(x, y)$ jesteśmy w stanie wyliczać w czasie stałym, możemy bez problemu utrzymywać wartość $hash(S)$.

Żeby S był zadanym obwarzankiem, musi zachodzić

$$hash(S) = (circHash(a, b, R) - circHash(a, b, L)) \mod p$$

gdzie $circleHash(a, b, r) = hash(\{(x, y) : dist((x, y), (a, b)) \leq r\})$.

Załóżmy, że znane nam są wartości $circleHash(0, 0, r)$ dla $r \in \{1, 2, ..., 5000\}$. Jeśli funkcja h będzie miała taką właność, że znając sumaryczny hash zbioru możemy wyznaczyć hash tegoż zbioru po przesunięciu jego wszystkich punktów o zadany wektor, to z $circleHash(0, 0, r)$ wyznaczymy $circleHash(a, b, r)$.

Czas wybrać funkcję h. Niech

$$h(x, y) = r^x s^y \mod p$$

gdzie r i s są stałymi wylosowanymi jednostajnie z $[1, p-1]$.

Po odpowiednim preprocessingu, $h(x, y)$ jestesmy w stanie liczyć w czasie stałym; ponadto, jeśli przesuniemy S o wektor (a, b), to $hash(S)$ zmieni się na $r^a s^b hash(S)$.

Podczas naszego rozwiązania założylismy, że dla każdego $r \in \{1, 2, ..., 5000\}$ znamy liczbę punktów kratowych w kole o promieniu r i środku w $(0, 0)$, oraz sumaryczny hash tych punktów.

Wartości te możemy policzyć na samym początku, wykonując preprocessing w czasie $\mathcal{O}(C^2)$.

Złożoność: $\mathcal{O}(C^2 + n \log C)$ (po drobnych optymalizacjach $\mathcal{O}(C^2 + n)$).

Dlaczego to działa?

- szansa, że dla dwóch różnych zbiorów A, B zajdzie $hash(A) = hash(B)$ jest rzędu $\mathcal{O}(\frac{C}{p})$ (z lematu Schwartza–Zippela)
- szansa, że zbiór S błędnie sklasyfikujemy jako obwarzanek, nie przekracza $\mathcal{O}(\frac{C^2}{p})$