Pierwiastki
[Square root decomposition]

Podstawa

Mamy ciąg liczb $A = (a_1, a_2, \dots, a_n)$. Mamy następujący problem: obsłużyć q zapytań postaci 1) podaj sumę na przedziale $[i, j]$: $1 \le i \le j \le n$, 2) zmień wartość na danym indeksie na inną, podaną. Problem można rozwiązać dzieląc ciąg na przedziały składowe: „pierwiastki".

21				17				20				13			
5	8	6	3	2	7	2	6	7	1	7	5	6	2	3	2

1)

21				15				20				13			
5	8	6	3	2	5	2	6	7	1	7	5	6	2	3	2

2)

21				15				20				13			
5	8	6	3	2	5	2	6	7	1	7	5	6	2	3	2

Mamy rozwiązanie w $O(q_1 * \sqrt{n} + q_2)$ z bardzo małą stałą.

Do kminienia

I. Dodawanie na przedziale; inne operacje
II. Implementacja

Podział na algorytm [Combining algorithms]

A	F	B	A
C	E	G	E
B	D	A	F
A	C	B	D

Mamy daną prostokątną tabelę $w \times h$ jakichś wartości (przypuśćmy, że liter, ale niech wielkość alfabetu $\sigma = O(wh)$). Dla każdej z występujących wartości chcemy znaleźć jaka jest minimalna odległość manhattańska ($|x_1 - x_2| + |y_1 + y_2|$) między dwoma takimi samymi wartościami.

Dwa algorytmy aby rozwiązać ten problem:

a) Niech ilość wystąpień litery l to c_l. Wtedy można po prostu dla wszystkich wartości, dla każdej możliwe pary sprawdzić odległość. Złożoność $O(c_l^2)$ na literę.
b) Dla każdej z liter puszczamy równoległego BFSa ze wszystkich wystąpień tej litery. Złożoność $O(wh)$ na literę.

Podzielmy litery na dwie grupy: wielką W ($c_l \geq k$), i małą V ($c_l < k$). Zauważmy, że $|W| \leq \frac{\sigma}{k}$. Użyjmy algorytmu b) aby rozpatrzyć grupę wielką, oraz a) aby rozpatrzyć grupę małą. Otrzymujemy złożoność:

$$O\left(|W|wh + \sum_{c \in V} c_l^2\right) = O\left(\frac{\sigma}{k}wh + whk\right) = O\left(wh\left(\frac{\sigma}{k} + k\right)\right) = O\left(wh\left(\frac{wh}{k} + k\right)\right)$$

Ta wartość jest minimalna dla $k = \sqrt{wh}$. Czyli złożoność to:

$$O\left(wh\left(\frac{wh}{\sqrt{wh}} + \sqrt{wh}\right)\right) = O\left(wh\,2\sqrt{wh}\right) = O\left(wh\sqrt{wh}\right)$$

Do kminienia

I. Rosyjscy oligarchowie. Dane jest n oligarchów i m firm. Każda firma ma swoją wartość v_i. Oligarchowie mają sumaryczną wartość swojego majątku, która jest równa sumie wartości firm których są właścicielami (jedna firma może mieć wielu właścicieli, ale nic to nie zmienia). Obsłuż q zapytań: 1) Oligarcha staje się właścicielem danej firmy (i zostaje nim do końca). 2) Wartość firmy zmienia się na inną podaną. 3) Podaj sumaryczny majątek danego oligarchy.

II. Kontenery z XXIV OI.

Algorytm Mo

4	2	5	4	2	4	3	3	4

→

4	2	5	4	2	4	3	3	4

Masz dane q zapytań offline o przedziały w ciągu, takich że różna kolejność zapytań na wejściu nie zmienia wyniku. Dodatkowo, mając dane rozwiązanie (i inne dane) dla danego przedziału, umiesz go przekształcić w inny usuwając/dodając po jednym elemencie z lewej/prawej strony (jedną taką operację wykonujesz w $O(z)$)

Algorytm Mo sortuje przedziały w taki sposób, że wykonasz co najwyżej $O((n+q)\sqrt{n})$ operacji (sumaryczna złożoność $O(q \log q + z(n+q)\sqrt{n})$).

```
# query.left // Q -> pierwiastek do którego należy query
# `//` to dzielenie z zaokrągleniem w dół
Q = floor(sqrt(n))
def mo_compare(lhs, rhs):
    if lhs.left // Q != rhs.left // Q:
        return lhs.left // Q < rhs.left // Q
    else:
        return lhs.right < rhs.right
```

Tym razem dzielimy zapytania na pierwiastki, a nie ciąg. i-ty pierwiastek zawiera zapytania których lewy koniec jest z przedziału $[(Q-1)i, Qi)$.

Do kminienia

I. Dowód złożoności (Hint: amortyzacja)

II. Triangulacja. Masz dane punkty wielokąta oraz jego triangulację – podział wielokąta na trójkąty, podana w formie indeksów wierzchołków wielokąta każdego z trójkątów. Każdy z trójkątów ma jakiś kolor. Ile co najwyżej cięć wielokąta możesz wykonać, takich że w żadnym momencie jakieś dwie części mają wspólne kolory? (Alternatywnie można rozwiązać trikiem na łączenie zbiorów. Hint: preorder/postorder z Megalopolis).

W tym wielokącie można wykonać tylko jedno cięcie (na odcinku 2 – 4).
Trójkąty to (1, 2, 5), (2, 4, 5), (2, 3, 4).