

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ФИНАНСОВ АНАЛИЗ НА АКЦИИ

Подзадача 1

Ще обхождаме всички подинтервали в заявките явно. Има различни варианти за търсене на минимуми в тях:

- търсене на минимум в отрязък за време, пропорционално на дължината му;
- сегментно дърво за минимум ;
- явно предварително пресмятане на всички минимуми ;
- разредени таблици.

Подзадача 2

Ще групираме заявките по k и ще решаваме задачата за всяка стойност на k отделно. За фиксирано k можем да намерим минимумите в подинтервали за $O(n)$. След това можем да отговаряме на заявките.

Подзадача 3

Ще намерим минимумите за всички подинтервали с дължина k за $O(n)$, след което ще пресметнем префиксните им суми, за да можем да отговаряме на заявките за $O(1)$.

Подзадачи 4 –5

Ако всички елементи на масива са равни на 1 или 2, тогава минимумът е равен на 2 само върху отрязъци, състоящи се изцяло от двойки. За всяка позиция i ще намерим максималното k , такова че интервалът $[i; i+k-1]$ се състои изцяло от 2.

Ще сортираме заявките по нарастване на k . За всяка позиция i ще поддържаме стойността на минимума върху отрязъка $[i; i+k-1]$. Действително, с увеличаване на k минимумът върху такъв отрязък най-много веднъж се променя от 2 на 1, и за всяко k можем да запазим списък от позиции, за които това ще се случи точно в този момент.

Тогава отговорът на текущата заявка е сумата върху някакъв отрязък от такива минимуми. За да я намерим ефективно, ще поддържаме минимумите в сегментно дърво или дърво на Фенуик.

Аналогично, ако елементите в масива не надвишават стойности на масива A , то при нарастване на k минимумът на интервала $[i; i+k-1]$ ще се променя не повече от $A-1$ пъти. Можем да намерим стойностите на k , при които това се случва, като обходим масива отдясно наляво и поддържаме стек от рекордни минимуми.

Подзадача 6

В решението на предишната подзадача поддържахме стойностите на минимумите върху подинтервали с дължина k . Нека забележим, че всеки елемент a_i е минимум на някакъв подотрязък от тези интервали (възможно е и на празен). За по-голяма яснота ще приемем, че ако $a_i = a_j$, то за по-малък се счита елементът с по-малък индекс.

Нека L и R са индексите на най-близкия до i елемент, по-малък от a_i , съответно вляво и вдясно (ако такъв не съществува, то $L=0$ или $R=n+1$). Тогава a_i ще бъде минимум на отрязъка $[L; r]$ тогава и само тогава, когато $L < l \leq i \leq r < R$.

За всяко k ще намерим броя на подинтервалите с дължина k , за които a_i е минимум:

- k , ако $k \leq \min(R-i, i-L)$
- $\min(R-i, i-L)$, ако $\min(R-i, i-L) \leq k \leq \max(R-i, i-L)$

- $R-L-k$, ако $\max(R-i, i-L) \leq k \leq R-L$
- 0, ако $k \geq R-L$

По този начин приносът на i -тия елемент към сумата от минимумите върху всички последователни подинтервали с дължина k (т.е. a_i , умножено по броя такива прозорци) е частично-линейна функция. За да намерим L и R за всички i , можем да използваме познат линейен алгоритъм със стек за рекордни елементи.

В подзадачата трябва да намерим стойността при фиксирано k на сумата от тези частично-линейни функции за всички i . Сумата от частично-линейни функции също е частично-линейна функция, която можем да получим, като съберем за всички i измененията на наклона във всяка точка k .

Подзадача 7 (пълно решение)

В пълното решение трябва да намерим сумата от минимумите върху някакъв подотрязък от интервали, т.е. да сумираме не всички частично-линейни функции, а само определен подотрязък от тях, като допълнително отчетем стойностите по краищата на отрязъка.

Нека разгледаме заявка (l, r, k) . Нека a_i е минимум на отрязъка $[l; l+k-1]$, а a_j е минимум на $[r-k+1; r]$. Ако $i = j$, тогава a_i е минимум на всички разглеждани подинтервали. Иначе a_i ще бъде минимум на $\min(i-l+1, R_i-l-k+1)$ подинтервала, а a_j ще бъде минимум на $\min(r-j-1, r-L_j-k-1)$ подинтервала.

Сега остава да сумираме частично-линейните функции за позициите от $i+1$ до $j-1$. Във всеки момент можем да ги поддържаме като линейни функции, като ги изменяме в точките $\min(R_i-i, i-L_i)$, $\max(R_i-i, i-L_i)$, R_i-L_i . След това, като построим сегментно дърво или дърво на Фенуик върху линейни функции, можем да намираме сумата им върху даден отрязък. Като заместим текущата стойност на k в получената линейна функция, получаваме приноса на елементите a_{i+1}, \dots, a_{j-1} .

Така получаваме пълно решение със сложност $O((n+q)\log n)$.

Автор: Кинка Кирилова-Лупанова