

Добуток

1. Будемо перебирати перший мінімум, другий мінімум і розширяти відрізок, сумарно це все працюватиме не довше $O(n^3)$.
2. Будемо перебирати початок відрізка, далі будемо розширювати його вправо підтримуючи в `std::multiset` множину елементів, для кожної правої границі візьмемо з мультисета два мінімуми та порахуємо добуток. Таким чином ми переберемо кожен відрізок і порахуємо відповідь за $O(n^2 \cdot \log(n))$.
3. Будемо робити все як в блоці 2, але мінімуми підтримуватимемо парою чисел.
4. Цей блок був створений для різноманітних евристик (переборів з відсіканням, чи чогось подібного).
5. Будемо перебирати другий мінімум. Для кожного числа порахуємо перше менше нього справа і зліва від нього - це можна зробити стеком. Потім для кожного другого мінімуму ми можемо взяти як перший мінімум або перше менше зліва, або справа, обидва взяти не можемо, візьмемо одне, а потім друге, опишемо процес з першим меншим зліва, справа буде симетрично. Нехай зараз ми розглядаємо i - як другий мінімум. l - індекс першого меншого зліва, r - справа. На цей час ми визначились, що точно візьмемо i та l , а отже відрізок $a[l, i]$. Тепер ми хочемо його максимально розширити так, щоб значення двох мінімумів не змінилось. Розширити вправо можна лише до r виключно, оскільки з визначення $a_i > a_r$. Розширити вліво можна до першого меншого за a_i виключно, серед тих, які лежать лівіше від позиції i , це ми можемо робити тримаючи останні дві позиції для кожного значення яке можуть приймати елементи масиву і будемо завдяки цьому шукати лівий кінець відрізка за $O(\sqrt{n})$ і тоді загальна складність алгоритму буде $O(n \cdot \sqrt{n})$.
6. Робитимемо все як в блоці 5, але пошук лівої границі можемо робити бін пошуком і `sparse table`, або деревом відрізків з каскадним спуском. Це працюватиме за $O(n \cdot \log(n))$.

Тут можна почитати як шукати найближчий менший елемент: <https://bit.ly/3XIVqIj>

Тут можна почитати про `sparse table`: bit.ly/3RcNFrm0

Тут можна почитати про дерево відрізків з каскадним спуском: bit.ly/3kKQ28J0