

Поиск пирамиды

Автор задачи: Владислав Власов, разработчики: Дмитрий Гнатюк и Даниил Орешников

Для решения задачи воспользуемся деревом отрезков. Заметим, что нам необходимо находить максимальный по длине отрезок на всем массиве, в котором элементы сначала возрастают, а затем убывают. Для подсчета соответствующего значения в вершине дерева отрезков давайте хранить 1. наибольший такой отрезок, 2. наибольший отрезок с началом в левой границе вершины, 3. наибольший отрезок с концом в правой границе вершины.

Для пересчета соответствующего значения в вершине через детей достаточно заметить, что ответ для нее либо полностью лежит в левом ребенке, либо полностью в правом, либо на стыке, но тогда и только тогда, когда

- либо слева в конце нет убывающей части и конец левого ребенка меньше начала правого
- либо справа в начале нет возрастающей части и конец левого ребенка больше начала правого
- либо левая половина заканчивается возрастающим отрезком, а правая — убывающим, тогда не важно в каком они отношении

Нетрудно заметить, что все эти величины можно пересчитывать за $\mathcal{O}(1)$, аккуратно разбирая случаи различного местоположения начала и конца ответа, а операция обновления в дереве отрезков будет работать за $\mathcal{O}(\log n)$, так как затронет только две вершины на нижнем уровне. Таким образом, мы получаем решение, работающее за $\mathcal{O}(\log n)$ на операцию.

Такой же асимптотики можно было добиться, храня описанные отрезки в куче (`std::set`), и проводя операции их склеивания и разделения при изменении элементов массива, однако такое решение может потребовать разбора порядка 37-и случаев.