Разбор задачи «Ёлки»

Решение задачи состоит в моделировании процесса, описанного в условии. Ясно, что после каждого дня количество ёлок уменьшится хотя бы на единицу; поэтому общее число промоделированных дней не превзойдёт исходного количества ёлок. Осталось только научиться быстро находить ёлку, пушистость которой минимальна, а при равенстве пушистостей — минимальную по номеру. Два возможных решения, позволяющие промоделировать один день за $O(\log n)$ операций — это использование двоичной кучи и построение дерева отрезков. Остановимся на первом способе как на наиболее естественном.

Наряду с пушистостью каждой ёлки будем хранить минимальную из пушистостей первой и второй ёлок, минимальную из пушистостей третьей и четвёртой и так далее (таких вспомогательных чисел будет $\frac{n}{2}$). Далее, по этим числам получим минимальную из пушистостей первых четырёх ёлок, минимальную из пушистостей ёлок 5, 6, 7, 8, и так далее (этих чисел будет уже $\frac{n}{4}$). Продолжив построение таким образом, мы в конце концов дойдём до уровня, на котором хранится одно число — минимальная изо всех пушистостей; при этом всего уровней будет $\log n$, а на k-ом из построенных уровней каждое число отвечает за отрезок из 2^k ёлок. Теперь, чтобы найти, какая же ёлка имеет эту минимальную пушистость, следует спуститься на предыдущий уровень и посмотреть, какое из двух чисел на нём равно общему минимуму, затем от него ещё на уровень ниже и так далее, пока мы не окажемся на уровне, где каждое число отвечает только за одну ёлку. Когда же мы изменим какую-то из пушистостей, это изменит не более чем $\log n$ хранимых чисел, поскольку изменённая ёлка содержится ровно в одном отрезке каждого уровня.

Подобную структуру данных удобно хранить как двоичную кучу. Число, отвечающее за минимум пушистостей всех ёлок, будем хранить в $a[1]$. Числа предыдущего уровня, отвечающие за левую и правую половины всего множества ёлок, запишем в $a[2]$ и $a[3]$, и так далее. Наконец, числа, соответствующие индивидуальным ёлкам, положим в $a[2^d]$, $a[2^d+1]$, $\ldots$, где $d = \log n$. Условимся обозначать несуществующие ёлки большой константой C. Наконец, дополним наши ёлки фиктивными (тоже с пушистостью C) так, чтобы слева от первой и справа от последней была по крайней мере одна ёлка и чтобы общее число ёлок равнялось 2^d. Теперь пересчитывать минимумы при изменении пушистости можно по простой формуле $a[i] = \min\{a[2 \cdot i], a[2 \cdot i + 1]\}$, а после нахождения наименее пушистой ёлки j изменять пушистость надо у ёлок $j-1$, j и $j+1$ независимо от их существования. Процесс прекращается, когда $a[1] \geqslant C$.

Следует также отметить, что при удвоении пушистости одной ёлки несколько раз пушистость может оказаться слишком большой, чтобы поместиться в целочисленный тип. Этого можно избежать, используя вещественный тип для хранения пушистостей; особенности хранения вещественных чисел таковы, что при удвоении числа потери точности не происходит, и все пушистости будут храниться и сравниваться без погрешностей.