

Условие

Научитесь вычислять любую ассоциативную функцию на пути в дереве за $\mathcal{O}(\log n)$.

Решение

Сделаем преподсчёт двоичных подъёмов вверх и вниз : `jmpup`, `jmpdown`, для каждого подъёма посчитаем ассоциативную функцию в соответствующую сторону, т.е. для случая, когда мы поднимаемся вверх и когда мы спускаемся вниз : `fup`, `fdown`. Это делается за $\mathcal{O}(n \log n)$, т.к. функция считается таким же ДП, как и сами подъёмы.

Чтобы найти путь, найдём LCA(начало пути, конец пути), после чего поднимемся от начала пути до LCA, считая ассоциативную функцию по подъёмам, после чего спустимся от LCA до конца пути, также считая ассоциативную функцию. Т.к. двоичных подъёмов/спусков до LCA $\mathcal{O}(\log n)$, итоговая ассимптотика $\mathcal{O}(\log n)$

```

1 calc_fun(f, start, end)
2     res = 0
3     lca = LCA(start, end)
4     D = d[start] - d[lca]
5     curr = start
6     for k = log n ... 0
7         if D >= 2 ** k
8             res = f(res, fup[curr, k])
9             curr = jmpup[curr, k]
10            D -= 2 ** k
11     D = d[end] - d[lca]
12     for k = log n ... 0
13         if D >= 2 ** k
14             res = f(res, fdown[curr, k])
15             curr = jmpdown[curr, k]
16             D -= 2 ** k
17     return res

```

`d` — массив глубин вершин, 0 — нейтральный элемент для операции.