1

Заведем ДО. В каждом узле будем хранить массив $dp[3][3]$, где $dp[i][j]$ – количество способов добраться из клетки $L+i$ в клетку $R-j$ ($dp[i][j]=0$ если $L+i$ или $R-j$ не принадлежат $[L,R]$ или содержат препятствия). В листах все значения кроме $dp[0][0]$ равны 0. $dp[0][0]=1$, если лист отвечает за клетку без препятствия. В остальных узлах мы можем посчитать значения по формуле: $t.dp[i][j] = l.dp[i][0] \cdot (r.dp[0][j] + r.dp[1][j] + r.dp[2][j]) + l.dp[i][1] \cdot (r.dp[0][j] + r.dp[1][j]) + l.dp[i][2] \cdot r.dp[0][j]$. При добавлении/удаление препятствия обновим значения на пути от соответсвующего листа до корня.

2

3

Преобразуем исходный массив так, чтобы его элементы принимали значения $[0,n)$ и если $x<y$ в исходном массиве, то $x'<y'$. Заведем массив p в котором для каждого числа сохраним индекс его предыдущего вхождения (-1 если вхождение первое). Тогда для отрезка $[L,R]$ количество уникальных элементов это количество $x \in p : x < L$. Заведем ДО, узел которого хранит количество $x : L \leq x \leq R$. С помощью него мы можем за $O(\log n)$ узнать количество добавленных элементов $< x$. Будем последовательно добавлять в него элементы массива p. В момент, когда мы добавили k элементов массива мы можем ответить на запрос для префикса массива длины k. Количество уникальных элементов на отрезке $[L,R]$ это количество элементов $< L$ на префиксе R минус количество таких элементов на префиксе $L-1$. Запомним все необходимые запросы и запомним ответы на них в процессе добавления элементов в ДО. Тогда после прохождения массива p мы будем знать ответы на все запросы.

4

Для прибавления x на пути $v \Rightarrow u$ прибавим x на пути $root \Rightarrow v$ и $root \Rightarrow u$ и $-x$ на пути $root \Rightarrow lca(u,v)$. Для прибавления на пути от корня до вершины используем Link-Cut Tree.