1 Задача 1

Построим ДО, в каждой вершине будем хранить матрицу ДП 3 на 3. Пусть $dp[i][j]$ — количество способов добраться из клетки с индексом $a+i$ до клетки с индексом $b-j$, где a и b —границы соответсвующей вершины ДО.

Пересчитывать значение в вершины v будем так: пусть a, b — границы левой вершины, c, d — границы правой вершины, пересчитаем для отрезка a, d. Так как кузнечик прыгает на 1, 2, 3, то он, очевидно, должен посетить одну из клеток $a+1$, $a+2$, $a+3$, и одну из $d-1$, $d-2$, $d-3$. Тогда $dp_{[a,d]}[i][j] = \sum_{t,k=1}^{3} dp_{[a,b]}[i][t] * dp_{[c,d]}[k][j]$ для $i, j \leq 3$(просто перебираем все способы).

Добавляем препятствие так — помечаем вершину как препятствие(количество способов в ней 0).

Ассимптотика $O(logn)$ на каждый запрос

Ответ, на запрос это функция на отрезке, объединяем по 2 как было указано выше для $[a,b], [c,d]$, результат будет лежать в $dp[0][0]$ соответвующего объединения.

2 Задача 3

Назовем исходный массив — a. Заведем массив $first$, где $first[i]$ будет равно 1, если число $a[i]$ встречается в массиве первый раз на позиции i, и 0 иначе. Построим дерево отрезков на этом массиве и будем считать в вершинах сумму.

Определим функцию $next(i)$ — она будет выдавать следующую позицию элемента равного $a[i]$. Реализовать можно так: заведем хешмап векторов, пройдемся по массиву a и будет класть все вхождения элемента в вектор. Тогда для запроса $(0, r)$ ответом будет, очевидно, сумма на подотрезке $(0, r)$ в ДО. Отсортируем все запросы по левой границе. Пусть l' — граница предыдущего запроса(для первого запроса $l' = 0$), l — граница текущего запроса, тогда в силу очевидной корректности алгоритма для запросов вида $(0, r)$, мы должны сделать значение массива как будто l это первый элемент массива, для этого пройдемся от l' до $l-1$ и если $first[i]$ равно 1 то делаем $set(next(a[i]), 1))$, где set — устанавливает значение листа в дереве отрезков. После этого берем сумму на отрезке (l, r) — это и есть ответ на запрос.

Время работы n запросов $O(nlogn)$: сортировка — $nlogn$, построение ДО — $nlogn$, прохождение по массиву и изменение элементов $nlogn$ и ответы на n запросов — $nlogn$

3 Задача 4

Присвоим всем вершинам дереве номера от 1 до n(обойдем в порядке дфса и будем присваивать вершине при выходе из нее следующий номер). Заметим что поддерево вершины — это отрезок.

Построим по этим номерам дерево отрезков, в котором будем хранить сумму. Когда нам пришел запрос на прибавление x на пути от u до v сделаем следующее:

1. Прибавим x к вершине u
2. Прибавим x к вершине v
3. Отнимем x от $lca(u, v)$
4. Отнимем x от $parent(lca(u, v))$

Теперь чтобы найти значение вершины t нужно взять сумму его поддерева(в ДО это сумма на соответсвующем отрезке).
Докажем, что это правда и прибавление на пути от u до v корректно, рассмотрим все возможные случаи:

1. t лежит в другом поддереве: ничего не измениться в сумме ее поддерева — верно
2. t лежит в поддереве $lca(u, v)$ но не на пути: ее сумма не изменится — верно
3. t это $lca(u, v)$: тогда в ее поддереве сумма претерпела следующие изменение $-x + x + x = x$ — верно
4. t выше $lca(u, v)$: $-x - x + x + x = 0$ — верно
5. t на пути от u до v: $+x$ — верно

Итоговая ассимтотипа $O(nlogn)$ — ДО плюс ДФС и $logn$ на каждый запрос