

1.

Для решения задачи будем использовать avl - дерево. В каждом узле, помимо информации о левом и правом ребенке, высоте дерева и номера комнаты, будем хранить размер поддерева с корнем в этой вершине.

Теперь, стоит заметить, что если обходить дерево в отсортированном порядке, то, ответ на запрос о текущей комнате будет как раз порядковый номер данной комнаты в обходе, т.е. равен числу комнат меньших данной. Тогда ключом будет являться как раз размер поддерева с корнем в данной вершине и мы, аналогично задаче в предыдущем дз(Stars) можем рассчитать порядковый номер.

но, такое решение в лоб оказывается неприемлемым из-за возможно большого количества узлов, в задаче гарантированно то, что количество запросов будет гораздо меньше числа комнат, а следовательно и количество запросов на удаление будет меньше, поэтому будем строить дерево реально удаленных вершин. Тогда, для удаления узла необходимо сначала найти реальную комнату и потом добавить эту вершину в дерево, соответственно пересчитывая соответствующие параметры для узла.

начнем обход с корня дерева удаленных вершин, и, будем его совершать, пока это возможно. тогда, когда мы стоим в корне, размер его левого поддерева - это количество удаленных вершин, меньших корня (по номеру комнаты). Тогда, если наш запрос меньше этого количества, то мы переходим на левую ветку, в противном случае переходим на правую - т.е. мы знаем, что у нас есть гарантированное число не удаленных вершин, вычитаем его из номера запроса (т.е. мы говорим, что такое-то количество не удаленных комнат мы уже нашли, но этого мало, надо искать еще). и запоминаем ключ текущего узла(фактический номер комнаты) и переходим в правую ветку, продолжая расчеты. Запоминаем номер текущего узла - так как мы перешли в правую ветку и нам помимо размера левого поддерева необходимо еще и вычитать количество элементов меньших данного.

и ответ формируется как сумма: разница между количеством уцелевших вершин в дереве и поступившим запросом и ключом последней подходящей нам вершины

при таком подходе, если дерево удаленных вершин пусто, то мы даем ответ за константу.

2.

б) Для подсчета суммы воспользуемся splay деревом по неявному ключу, т.е. во-первых дополнительно в каждом узле будем хранить информацию о количестве элементов в левом поддереве с корнем в данной вершине, тогда обращаться к каждому элементу мы можем по номеру в порядке обхода(при переходе по левой ветке это количество соответственно равно количеству элементов в левом поддереве т.е. то, что мы храним, а при переходе направо необходимо это количество накапливать). во-вторых, в каждом узле будем дополнительно хранить информацию о сумме элементов в поддереве с корнем в данной вершине. Теперь надо найти вершины a и b, сделать split отдельно по каждой из вершин, и тогда в корне нового дерева будет храниться информация о сумме элементов от a до b.