4.1. Покраска на отрезке.

Основная идея уже рассказывалась нам на практике: запаздывающее обновление.

Заведем дерево отрезков и будет хранить для отрезка тот цвет, в который он полностью покрашен.

При операции обновления некого узла A нужно сначала «протолкнуть» информацию о цвете в дочерние узлы, а затем обработать этот узел A (вызывать функцию update либо от левого, либо от правого потомка).

Получение цвета происходит аналогично – проталкиваем цвет из узла до его потомков, и рекурсивно вызываемся из одного из потомков.

Ясно, что эта операция никак не изменяет асимптотику обычных операций на дереве отрезков, скажем, на дереве отрезков для суммы.

Ответ за запрос «покраска всего массива» работает аналогично: проталкивание + обход дерева.

4.2. Найти первый меньший элемент справа от заданного

Заведем дерево отрезков и будем для каждого отрезка хранить значение минимума на нем. Обновление такого дерева мало отличается от обновления обычного дерева отрезков для суммы.

Применяя аналогичные лекционным рассуждения к этой задаче, ее можно свести к такой: дан отрезок и число A, нужно найти на этом отрезке первый элемент, меньший заданного числа A. Почему так можно сделать? Ясно, что нужный отрезок либо разбивается серединой на два (один из которых сразу подходит под новое рассуждение), либо полностью лежит в одной из половинок: а поэтому сводится к случаю один.

Итак, если научться решать новую задачу, исходная решается просто: ищем ответ на обоих (если их два) подотрезках и выводим тот, что ближе к заданному числу A.

А задача на всем отрезке решается просто: так как мы знаем значения минимума на каждой половинке отрезка, то мы можем рекурсивно запуститься на одной из них простым разбором случаев: если оба минимума меньше A, то запускаться стоит на левом отрезке, если ровно один меньше: то на том, где этот минимум меньше. Если же оба минимума больше A, то на этом отрезке такого числа нет.