После того, как фокусник Дормидонт произвел фурор своим новым карточным фокусом с пассами и угадыванием позиций карт, директор цирка попросил его изучить еще пару фокусов такого же типа, пообещав повысить зарплату вдвое. Дормидонт, конечно, согласился, и предложил следующий вариант фокуса:

1) n карт раскладывается в ряд по одной карте (рубашками вверх). Пусть pk – позиция k-й слева карты в ряду, то есть p1 — позиция самой левой карты в ряду, а pn — позиция самой правой;

2) Фокусник делает m пассов, в результате которых некоторые карты перемещаются на новые позиции. Каждый пасс характеризуется двумя параметрами: l и r. Эти параметры означают, что все карты, находящиеся на позициях с pl по pr включительно, перемещаются в начало с сохранением их относительного порядка, а карты, находившиеся на позициях с p1 по pl-1 включительно, если такие есть, перемещаются вправо на r-l+1 позицию, занимая таким образом все освободившиеся позиции.

3) После того, как фокусник сделал все пассы, он должен уметь угадывать, на какой позиции находится любая из карт.

Пока Дормидонт разучивает новые пассы, помогите ему научиться определять конечный порядок карт.

В первой строке два целых числа n и m (2 ≤ n ≤ 100 000, 1 ≤ m ≤ 100 000) — количество карт и количество пассов.

Следующие m строк содержат по два целых числа li и ri (1 ≤ li ≤ ri ≤ n).

Выведите n целых чисел — порядок карт в конечном положении после выполнения всех пассов.

**Решение:**

Заметим, что один пасс являет собой перемещение отрезка в начало массива. Именно поэтому можно представить, что вы из массива как буд-то вырезаем этот отрезок, переставляем его в начало, а потом объединяем эти отрезки снова в массив.

Данные операции на массиве может реализовывать структура данных “Декартово дерево по неявному ключу” (ДД).

Если в качестве приоритета брать случайные значения, то средняя высота дерева будет O(logn), где n – количество вершин. Доказательство можно увидеть в “**Proof\_of\_treaps\_height.docx**”.  
А, так как операции **split** и **merge** полностью зависят от высоты дерева, то и асимптотическое время работы каждой из них будет составлять O(logn).

Операция пасса будет происходить следующим образом: ДД разрезают двумя операциями **split** по ключам (**l – 1**) и (**r**) на части **left, mid, right**. Потом **left** и **mid** меняют местами. Затем объединяют данные ДД в одно 2мя операциями **merge**.

Что бы вывести массив, нужно выводить значения узлов по возрастанию ключа. Для этого нужно сначала вывести результат левого поддерева, затем само значение узла, и, наконец, результат правого поддерева.

Построить ДД можно за O(n \* logn) или O(n).  
Так как операция пасса зависит лишь от операций **split** и **merge**, то и асимптотика операции также будет O(logn). А суммарная асимптотика всех операций пасса – O(m \* logn).  
Операция вывода будет происходить за O(n).

Таким образом, суммарная временная асимптотика – O((m + n) \* logn) или O(m \* logn + n), что зависит от метода построения дерева.  
А помяти выделяется .