Деревья поиска Храним структуру данных, которая должна уметь делать все то же самое, что можно делать с отсортированным массивом, но еще с добавлением-удалением.

Binaary search tree (BST) Корневое дерево. В вершине храним левого сына, правого сына, предка, ключ.

Обозначения:

- $l(v)$ — левый сын
- $r(v)$ — правый сын
- $p(v)$ — предок
- $key(v)$ — ключ
- $T(v)$ — поддерево
- $S(v) = |T(v)|$ — размер
- $A(v)$ — множество предков
- $seg(v) = [\min_{u \in T(v)} key(u); \max_{u \in T(v)} key(u)]$

Условие BST:

$$\forall u \in T(l(v)) : \ key(u) < key(v)$$

$$\forall u \in T(r(v)) : \ key(u) > key(v)$$

То есть, ключи лежат в порядке лево-правого обхода (в порядке «выписать левое поддерево - выписать вершину - выписать правое поддерево»)

$v \in seg(u) \leftrightarrow v \in T(u)$

Поиск в дереве — надо сделать спуск. То есть, если текущая вершина — не та, которая нам нужна, то можно понять, где лежит нужная нам, с помощью условия BST.

Нахождение следующего — либо спуск вправо, либо подъем по предкам до первого большего.

Основная проблема — операции вставки/удаления, которые должны делать дерево сбалансированным (таким, высота которого нас устраивает, то есть примерно $\log n$)

Декартово дерево У каждой вершины будем хранить не только ключ, но и какой-то приоритет y. Построим дерево так, что по y это куча, а по x это BST.

Тогда декартово дерево задается однозначно, если определить все приоритеты. Почему? Расставим точки на плоскости в соответствии с (x, y), после чего найдем корень. У корня будет наименьший приоритет и поэтому он определяется единственным образом (будем считать, что приоритеты различны). Тогда к корню нужно приписать слева и справа по дереву, которые рекурсивно строятся в левой и правых частях.

Утверждение Если взять случайные y, то матожидание глубины дерева $O(\log n)$

Доказательство: Нет.

Утверждение Если взять случайные y, то матожидание глубины каждой вершины $O(\log n)$

Доказательство. Матожидание высоты вершины — это число вершин, которые являются ее предками. Вершина i будет предком вершины j, если $y_i = \max\{y_i, y_{i+1}, \ldots, y_j\}$. Матожидание суммы таких величин для i это $\sum\limits_{j=0}^{i-1} \frac{1}{i-j} + \sum\limits_{j=i+1}^{n-1} \frac{1}{j-i} \leq 2\sum\limits_{i=1}^{n} \frac{1}{i} = O(\log n)$