

## Содержание

1. BST .....	1
1.1. За $\mathcal{O}(1)$ .....	1
1.2. AVL .....	1
2. Хеш-Таблицы .....	2
2.1. Открытая адресация .....	2

## 1. BST

**Доделать**

Примечание:  $\text{next}(v)$  — спускаемся максимально влево

### 1.1. За $\mathcal{O}(1)$

Можем делать Find за  $\mathcal{O}(1)$  храня мапу из значений в вершину. Для минимума/максимуму храним соответственно ссылки на них.

Как делать Next за  $\mathcal{O}(1)$ : храним для каждой вершины Next и Prev. Обновляем при изменении, получается двусвязный список.

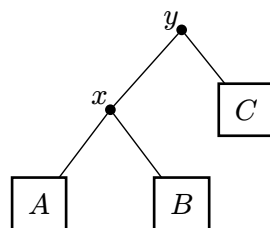
Примечание: Нельзя сделать Add за  $\mathcal{O}(1)$ , иначе смогли бы отсортировать массив за  $\mathcal{O}(n)$ .

### 1.2. AVL

**Определение 1.2.1:** Дерево обладает свойством AVL если для любой вершины, глубина левого поддерева отличается от глубины правого не более чем на 1.

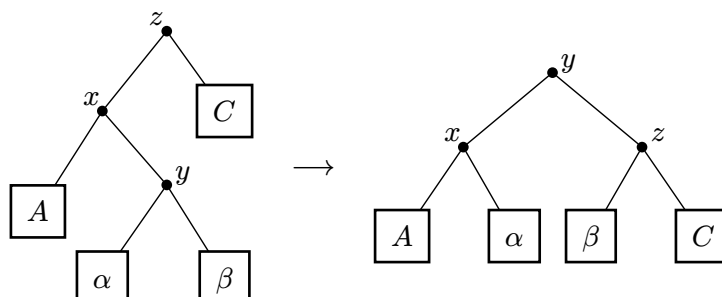
При добавлении поднимаемся на верх и делаем повороты.

Примечание:



Если после добавления в поддерево B получились деревья  $h_A = h$ ,  $h_B = h + 1$ ,  $h_C = h$ , то  $h_x = h + 2$  и нужно сделать поворот. Однако после поворота проблема останется.

Выделить корень z в поддереве B, и вынести его наверх, x слева, y справа — **большое вращение**.



**Теорема 1.2.1:** Всегда делаем не больше одного вращения

Примечание: При удалении можем пометить вершину как удаленную и что-то делать потом с ней **Доделать**

## 2. Хеш-Таблицы

Заводим массив массивов и мапим индексы в нем  $i \rightarrow i \bmod N$ , где  $N$  размер массива.

Примечание: Вместо фиксированного  $N$  берем рандом от  $N$  до  $2N$ . Тогда вероятность коллизии  $i \bmod N' \neq j \bmod N' \cdot i - j \mid N'$  с вероятностью(?)  $\log$ .

**Доделать**

### 2.1. Открытая адресация

Примечание: При удалении помечаем как “удаленный”