

---

АДРИАНОВ Н.М.  
ИВАНОВ А.Б.

# АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

ДВОИЧНЫЕ ДЕРЕВЬЯ ПОИСКА

# AVL ДЕРЕВЬЯ

1962 г. - Георгий Максимович Адельсон-Вельский, Евгений Михайлович Ландис

Инвариант: Для любого узла разница между высотами левого и правого поддерева не превосходит 1.

**Утв.** Высота AVL дерева -  $O(\log n)$

**Доказательство** Пусть  $N_h$  - минимальное количество вершин в AVL дереве высоты  $h$ . Чему равно  $N_h$ ?

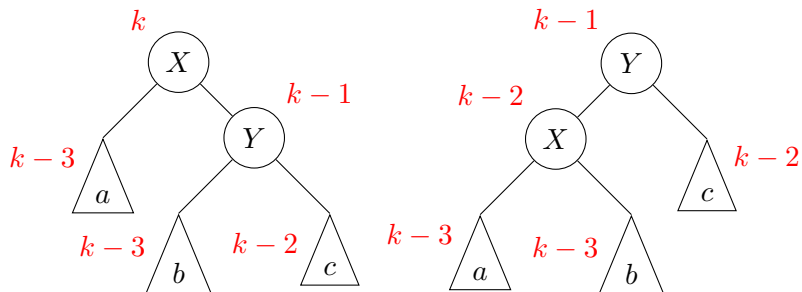
## AVL ДЕРЕВЬЯ, ВСТАВКА

Для восстановления свойства AVL дерева после вставки используются уже известные нам операции - поворот влево и поворот вправо.

Пусть  $X$  самый "нижний" узел, в котором нарушено свойство пусть правое поддерево  $x$  выше левого. Вершина правого поддерева -  $Y$ .

# AVL-ДЕРЕВО ВСТАВКА

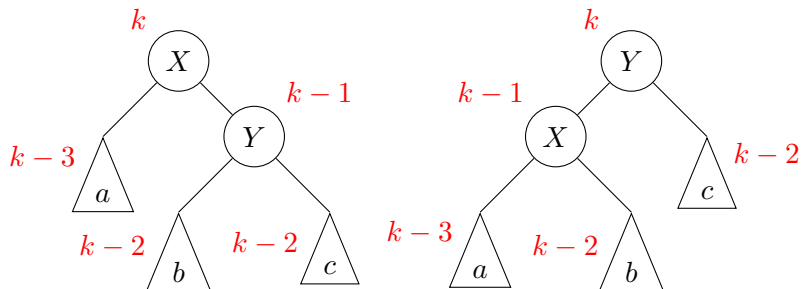
Случай 1: правое поддерево  $Y$  выше левого.



Выполняем левый поворот.

# AVL-ДЕРЕВО ВСТАВКА

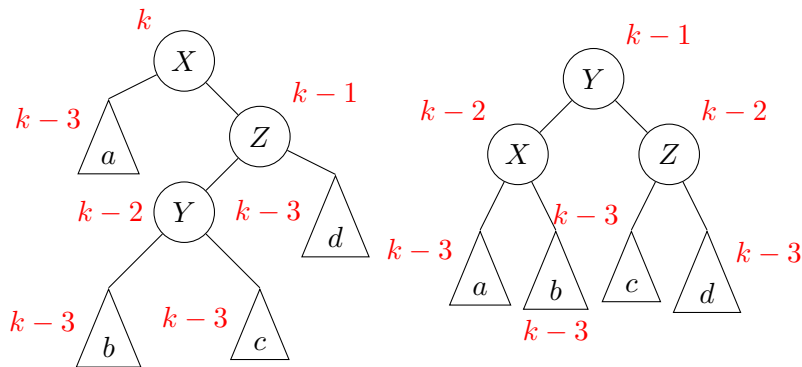
Случай 2: поддеревья  $Y$  одинаковой высоты.



Выполняем левый поворот.

# AVL-ДЕРЕВО ВСТАВКА

Случай 3: левое поддерево  $Y$  выше правого.

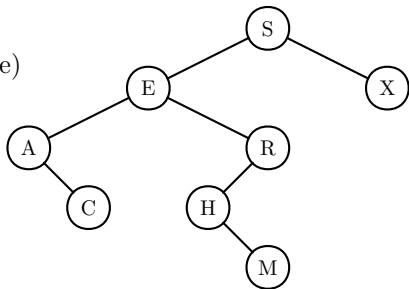


Выполняем правый поворот. Затем выполняем левый поворот. (Есть еще несколько случаев в зависимости от балансировки дерева с корнем  $Y$ .)

## ОПЕРАЦИИ В BST, ИСПОЛЬЗУЮЩИЕ ПОРЯДОК

В силу использования сравнений – у нас есть дополнительный набор операций, использующих порядок (не входящих в интерфейс ассоциативного массива)

- `min / max`
- `floor / ceiling`
- `rank`
- `in-order traversal`
- поиск по диапазону (`range`)



# ГЕОМЕТРИЧЕСКИЕ ПРИЛОЖЕНИЯ

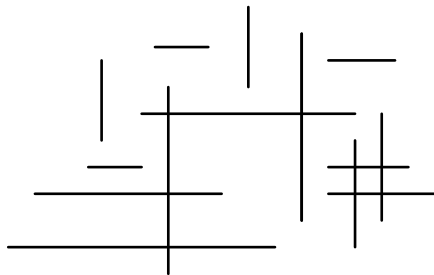
- Пересечения перпендикулярных отрезков
- Поиск в 2-мерном диапазоне
- Поиск ближайшего соседа
- Интервальное дерево поиска
- Пересечения перпендикулярных прямоугольников



# ПЕРЕСЕЧЕНИЯ ПЕРПЕНДИКУЛЯРНЫХ ОТРЕЗКОВ

**Дано:**  $N$  горизонтальных и вертикальных отрезков.

**Найти:** все точки пересечения.



## ПОИСК В 2-МЕРНОМ ДИАПАЗОНЕ

**Дано:**  $N$  точек на плоскости.

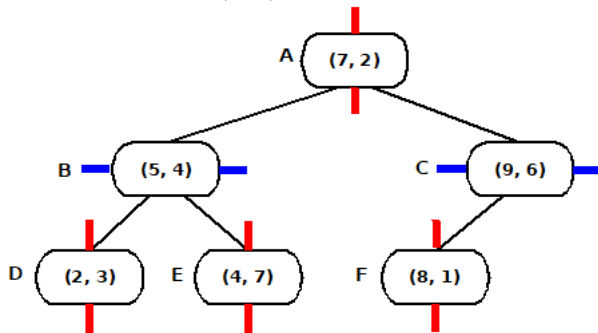
**Запрос:** Для прямоугольника  $[x_1, x_2] \times [y_1, y_2]$  найти все точки, лежащие в нем.

## 2D-ДЕРЕВО

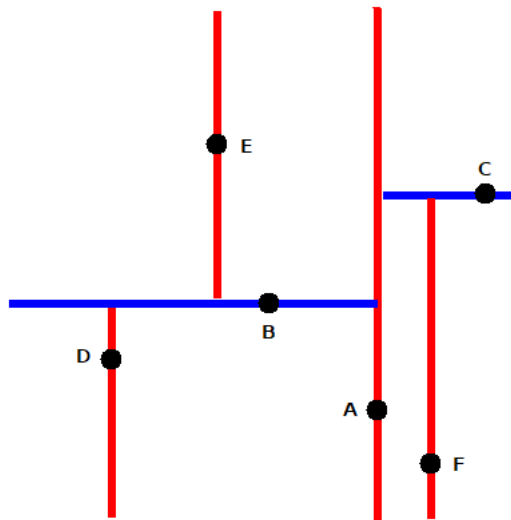
В вершинах храним пары ключей  $(a, b)$ .

В вершинах на уровнях 0, 2, 4, ... делим по координате  $x$ : в левом поддереве точки  $(x, y)$  с  $x$ -координатой  $x < a$ , в правом –  $x \geq a$ .

В вершинах на уровнях 1, 3, 5, ... делим по координате  $y$ : в левом поддереве точки  $(x, y)$  с  $y$ -координатой  $y < b$ , в правом –  $y \geq b$ .



## 2D-ДЕРЕВО



# ПОИСК БЛИЖАЙШЕГО СОСЕДА

**Дано:**  $N$  точек на плоскости.

**Запрос:** Для точки  $(x, y)$  найти ближайшую точку из этих  $N$ .

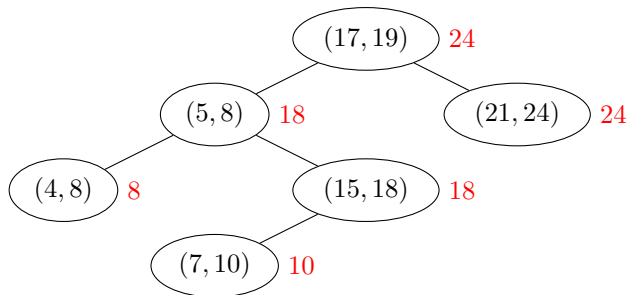
# ИНТЕРВАЛЬНОЕ ДЕРЕВО ПОИСКА

Структура данных для хранения (перекрывающихся) интервалов.

- Добавить интервал  $(a, b)$
- Найти интервал  $(a, b)$
- Удалить интервал  $(a, b)$
- Для заданного интервала  $(a, b)$  найти все пересекающиеся с ним интервалы, хранящиеся в нашей структуре.

# ИНТЕРВАЛЬНОЕ ДЕРЕВО ПОИСКА

- Левый конец интервала используем как ключ
- Дополнительно храним максимальный правый конец в поддереве



# ИНТЕРВАЛЬНОЕ ДЕРЕВО ПОИСКА

Алгоритм поиска интервала в интервальном дереве, пересекающегося с  $(a, b)$ :

- Если текущий узел пересекается с  $(a, b)$ , возвращаем его
- Иначе, если левое поддерево пустое, идем направо
- Иначе, если максимальный правый конец в левом поддереве меньше  $a$ , идем направо
- Иначе, идем налево



# ПЕРЕСЕЧЕНИЯ ПРЯМОУГОЛЬНИКОВ

**Дано:**  $N$  прямоугольников  $[x_1, x_2] \times [y_1, y_2]$ .

**Найти:** все пары пересекающихся прямоугольников.