

Декартово дерево

Алгоритмы 15 - Айрат Хасьянов

Treap = Дерамиды

- Структура данных, объединяющая бинарное дерево поиска и пирамиду (heap)
- Позволяет реализовать быстрые операции над массивами

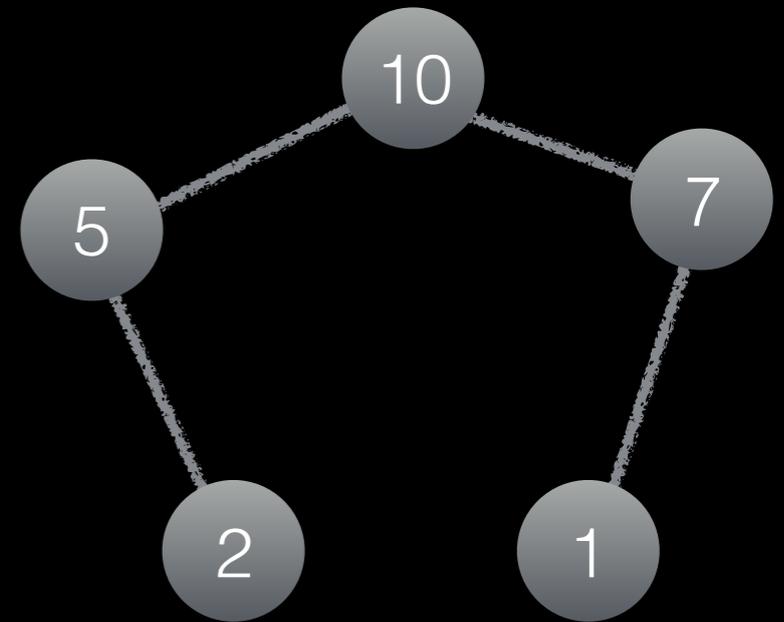
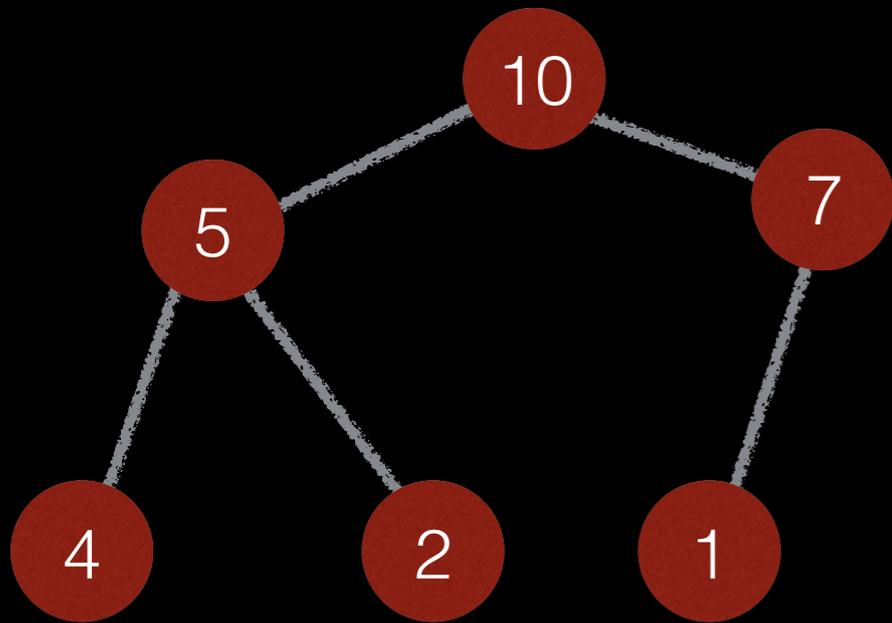
Вспомним пирамиду



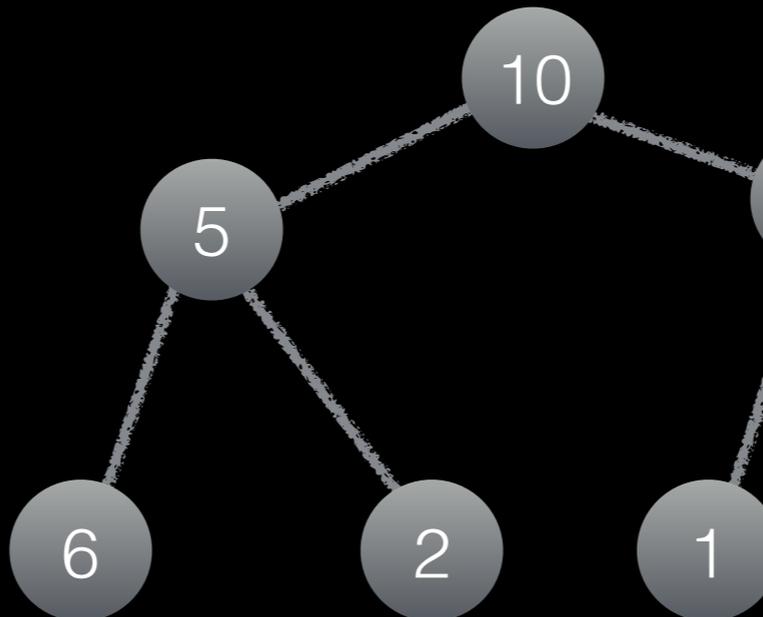
Пирамида

- Пирамида - это бинарное дерево с ключами, назначенными ее узлам (по одному на узел), что
 - А. (Форма) Бинарное дерево практически полное или просто полное - все уровни заполнены, только некоторые крайние справа листья могут отсутствовать;
 - В. (Доминирование) Ключ в каждом узле не меньше ключей в его дочерних узлах (за исключением листьев, конечно).

Пирамиды?

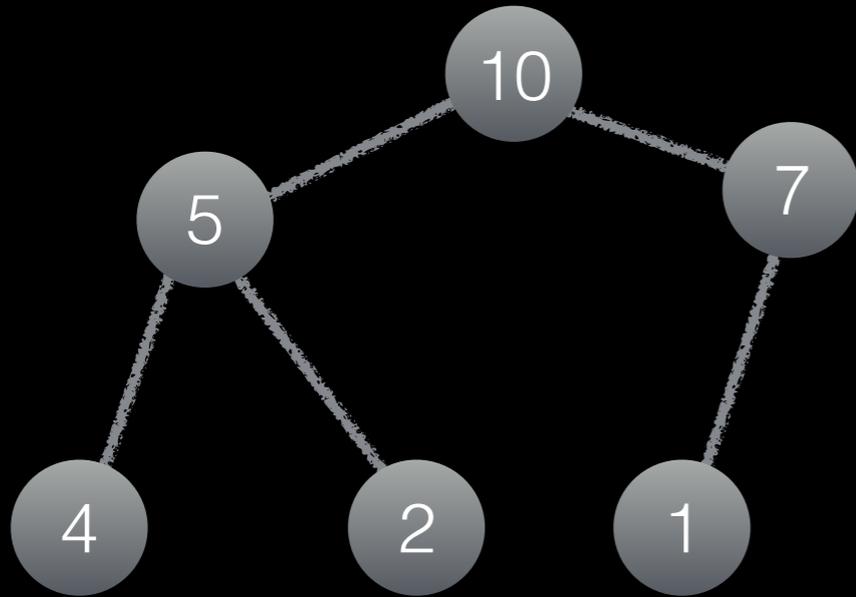


Форма



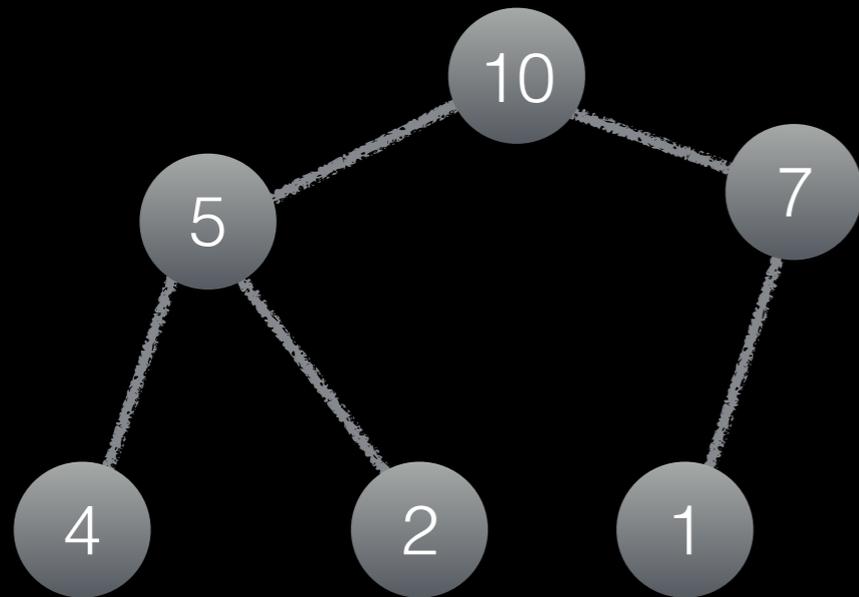
Доминирование

Свойства Пирамиды



1. \exists ровно 1 практически полное бинарное дерево с n узлами. Его высота $\lfloor \log_2 n \rfloor$.
2. В корне всегда наибольший элемент.
3. \forall узел пирамиды с потомками - тоже пирамида!

Представление Пирамиды

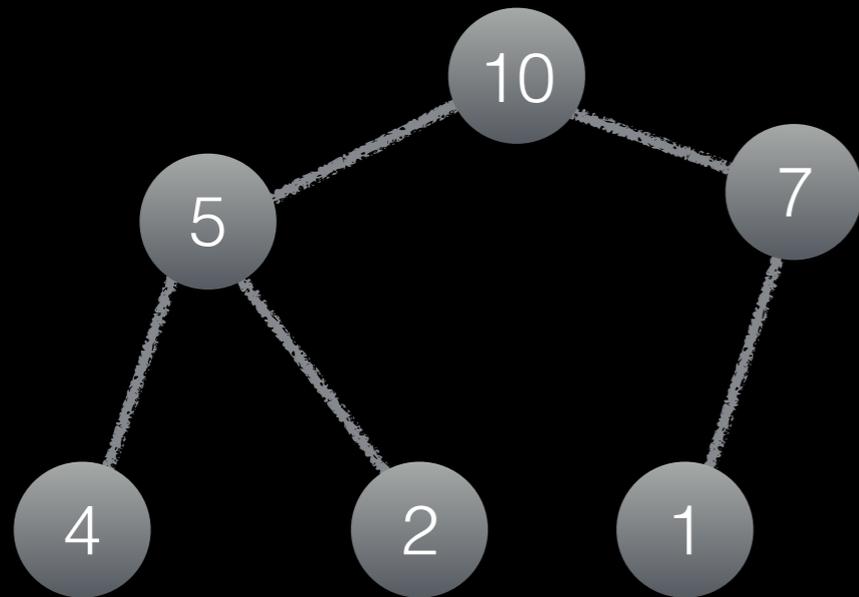


0	1	2	3	4	5	6
	10	5	7	4	2	1
	родительские			ЛИСТЬЯ		

Пирамиду можно представить массивом, записывая элементы сверху вниз, слева направо:

- А. ключи родительских узлов занимают первые $\lfloor n/2 \rfloor$ позиций, а ключи листьев - последние $\lceil n/2 \rceil$
- В. дочерние ключи по отношению к родительскому в позиции i находятся в позициях $2i$ и $2i+1$
- С. родительский ключ для ключа в позиции i ($2 \leq i \leq n$) находится в позиции $\lfloor i/2 \rfloor$

Пирамида как массив

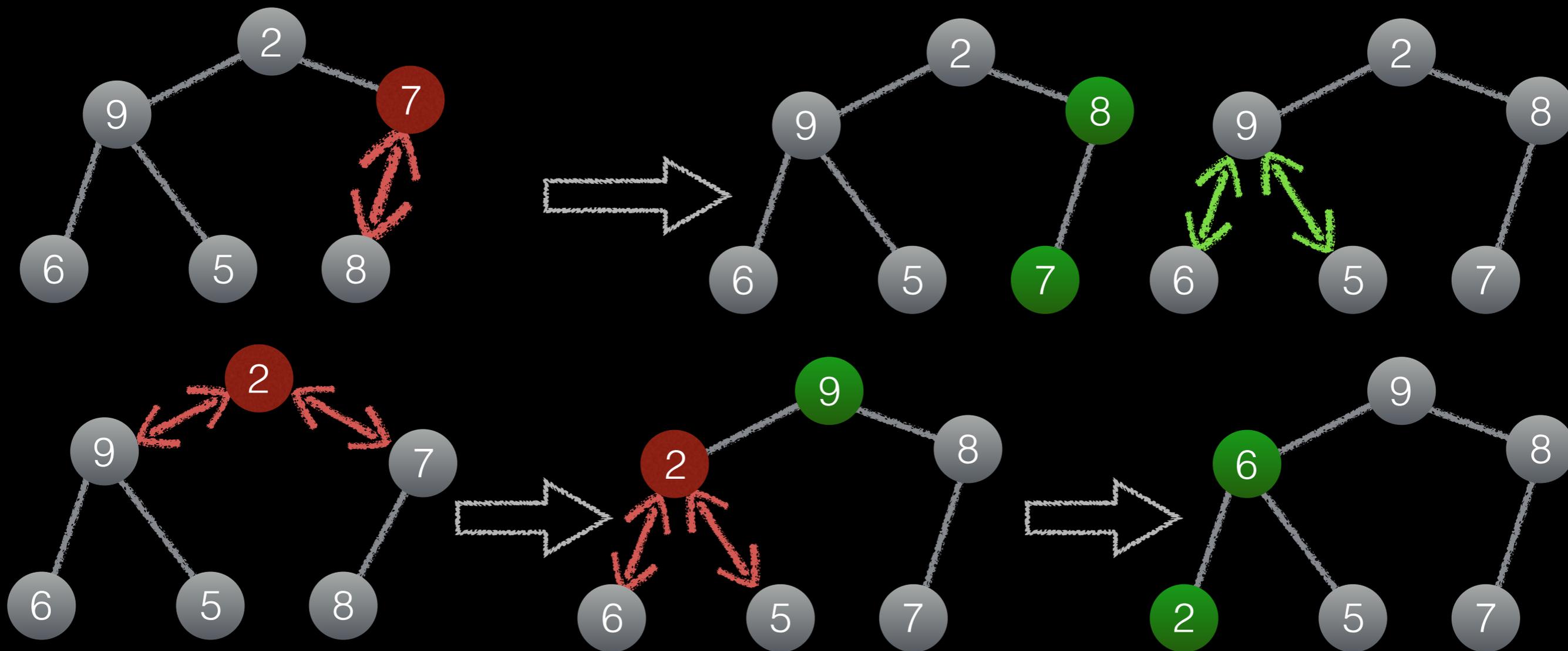


0	1	2	3	4	5	6
	10	5	7	4	2	1
	родительские			ЛИСТЬЯ		

Пирамида = массив $H[1..n]$, \forall элемент в позиции i в первой половине массива больше или равен элементам в позициях $2i$ и $2i+1$

Восходящее построение пирамиды

0	1	2	3	4	5	6
	9	2	7	6	5	8
	родительски			ЛИСТЬЯ		

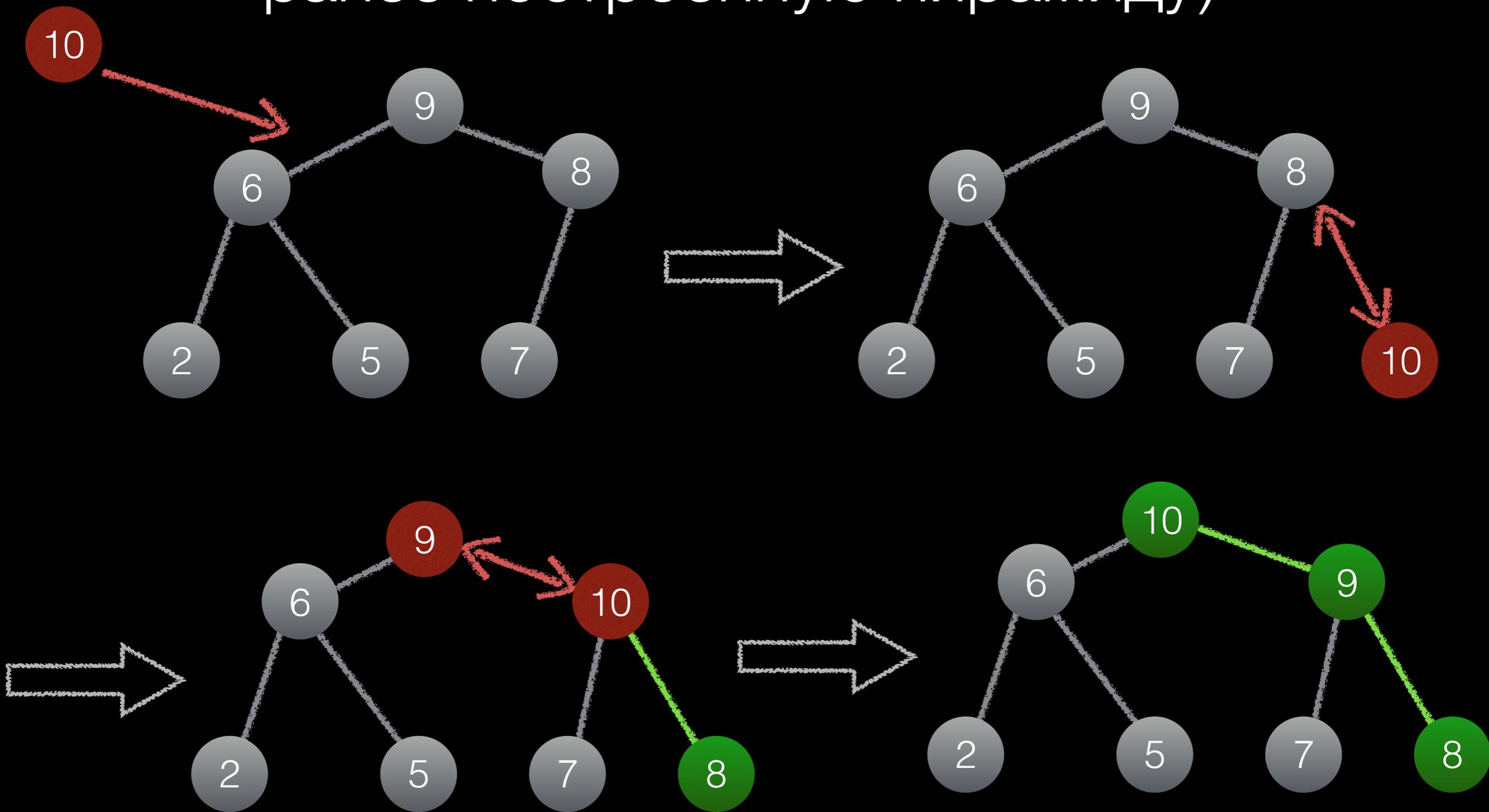


Постройте пирамиду для массива длиннее 10

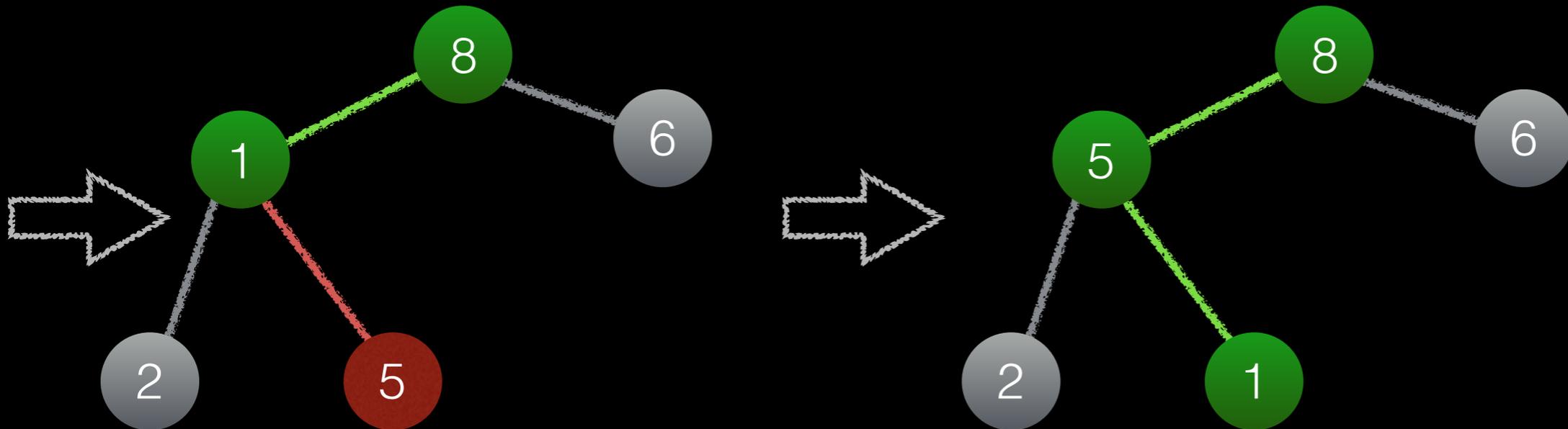
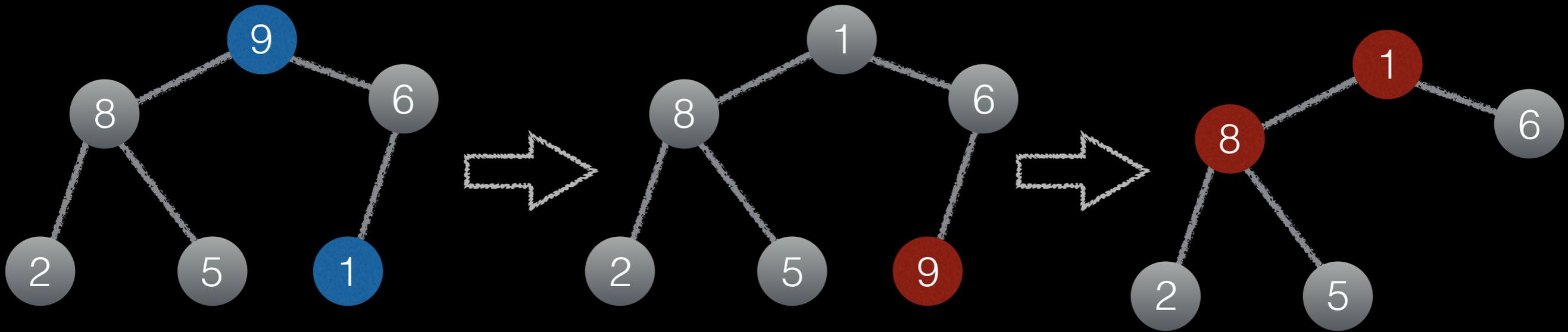
0	1	2	3	4	5	6	7	8	9	10	11	12
	9	2	7	6	5	8	1	3	10	11	4	12
							Листья					

0	1	2	3	4	5	6
	9	2	7	6	5	8
				Листья		

Нисходящее построение пирамиды (последовательная вставка ключа а ранее построенную пирамиду)



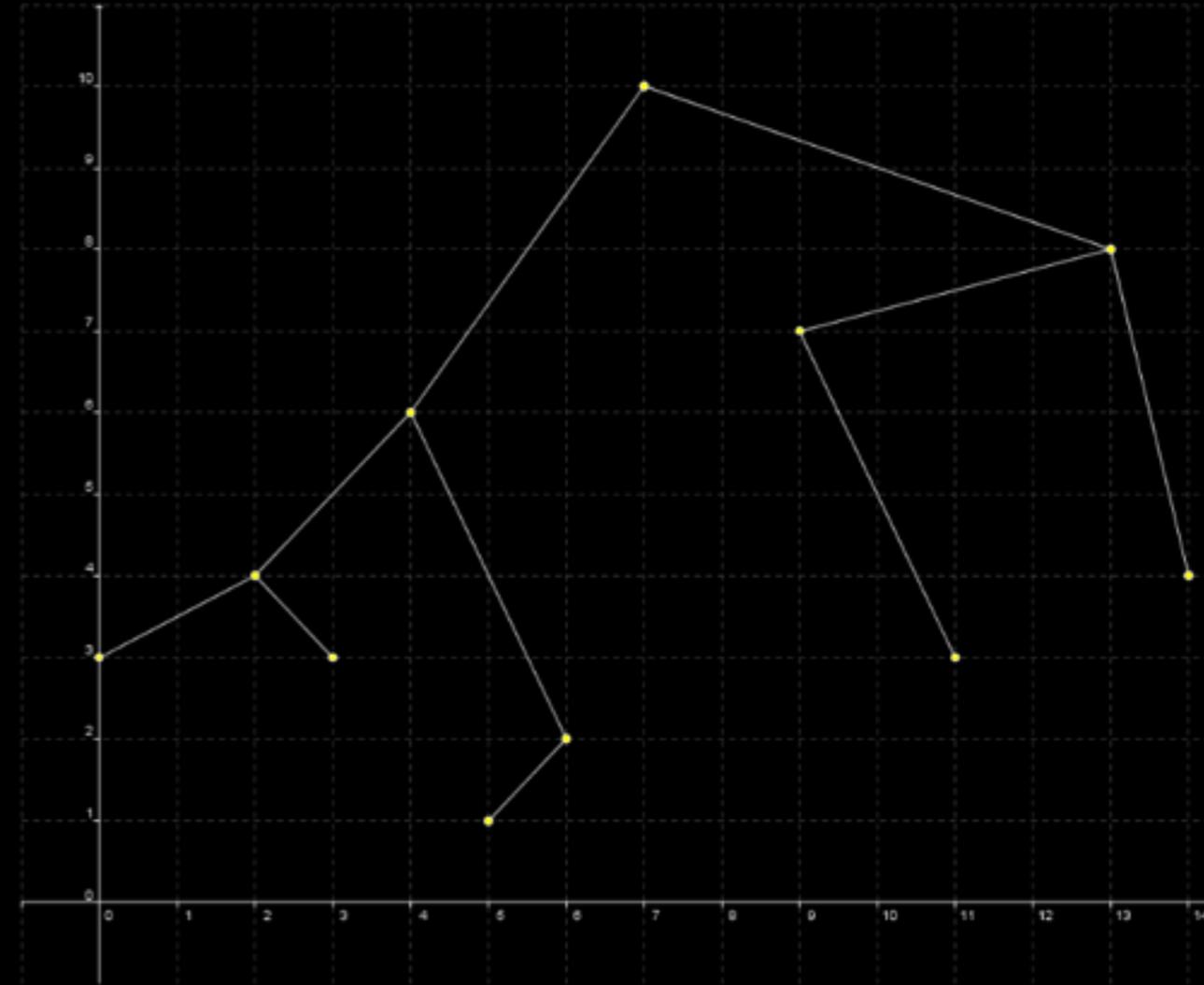
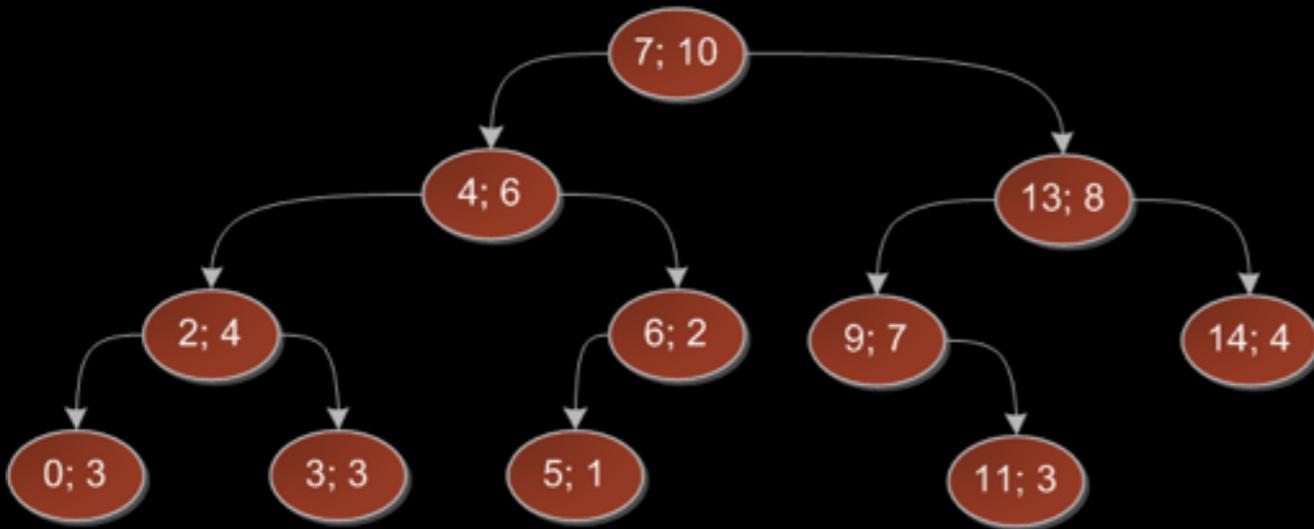
Удаление корневого узла из пирамиды



Итак, дерамида

- В каждом узле храним ключ x и приоритет y
- Так что узлы дерева = координаты на декартовой плоскости (x, y)
- Сколько деревьев с заданными приоритетами узлов можно портить?

Пример



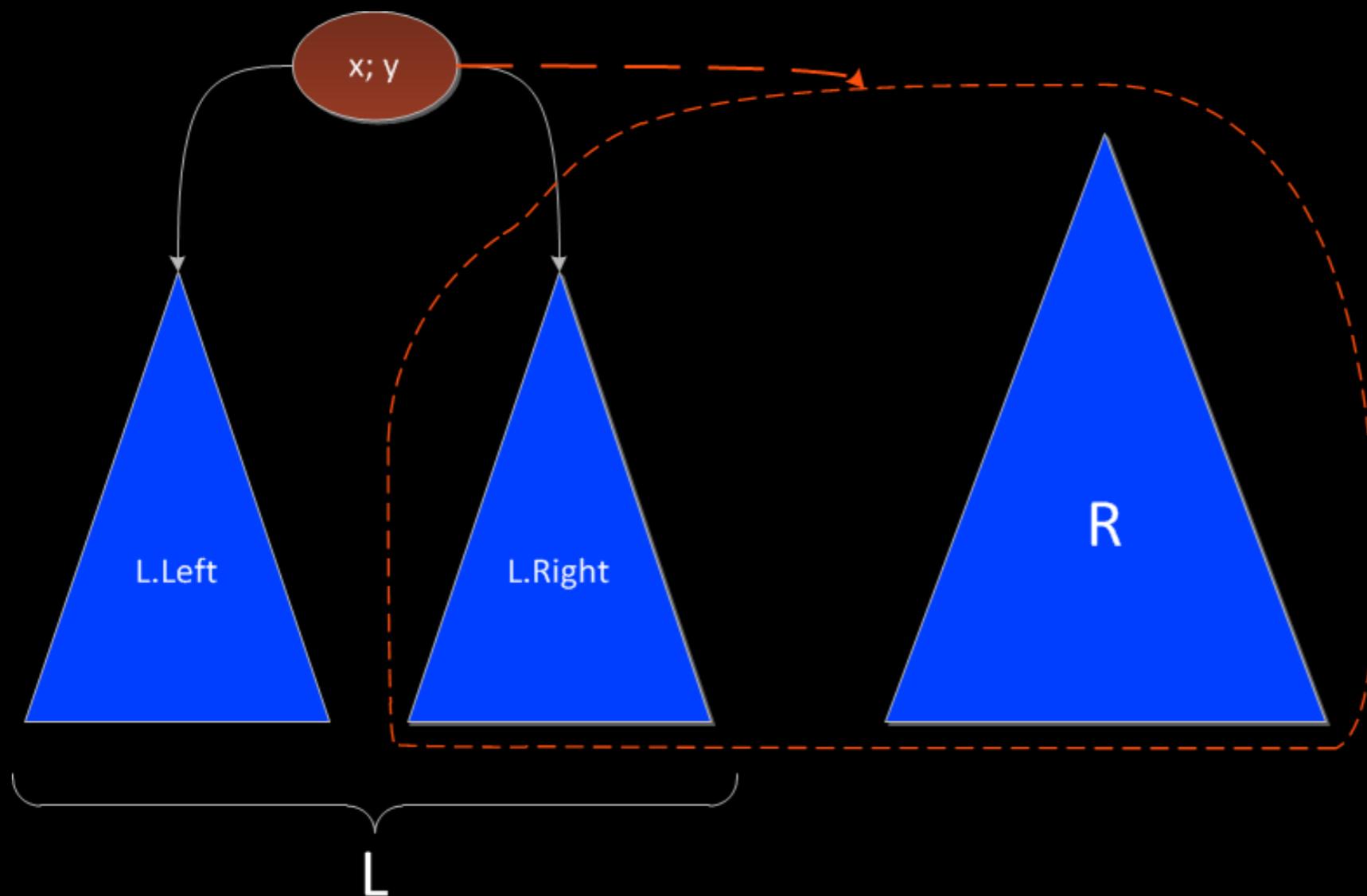
Здесь и далее <https://habrahabr.ru/post/101818/>

Сбалансированность

- Какова будет высота дерева, при случайно выбранных приоритетах узлов?
- Будет стремиться к $O(\log_2 n)$

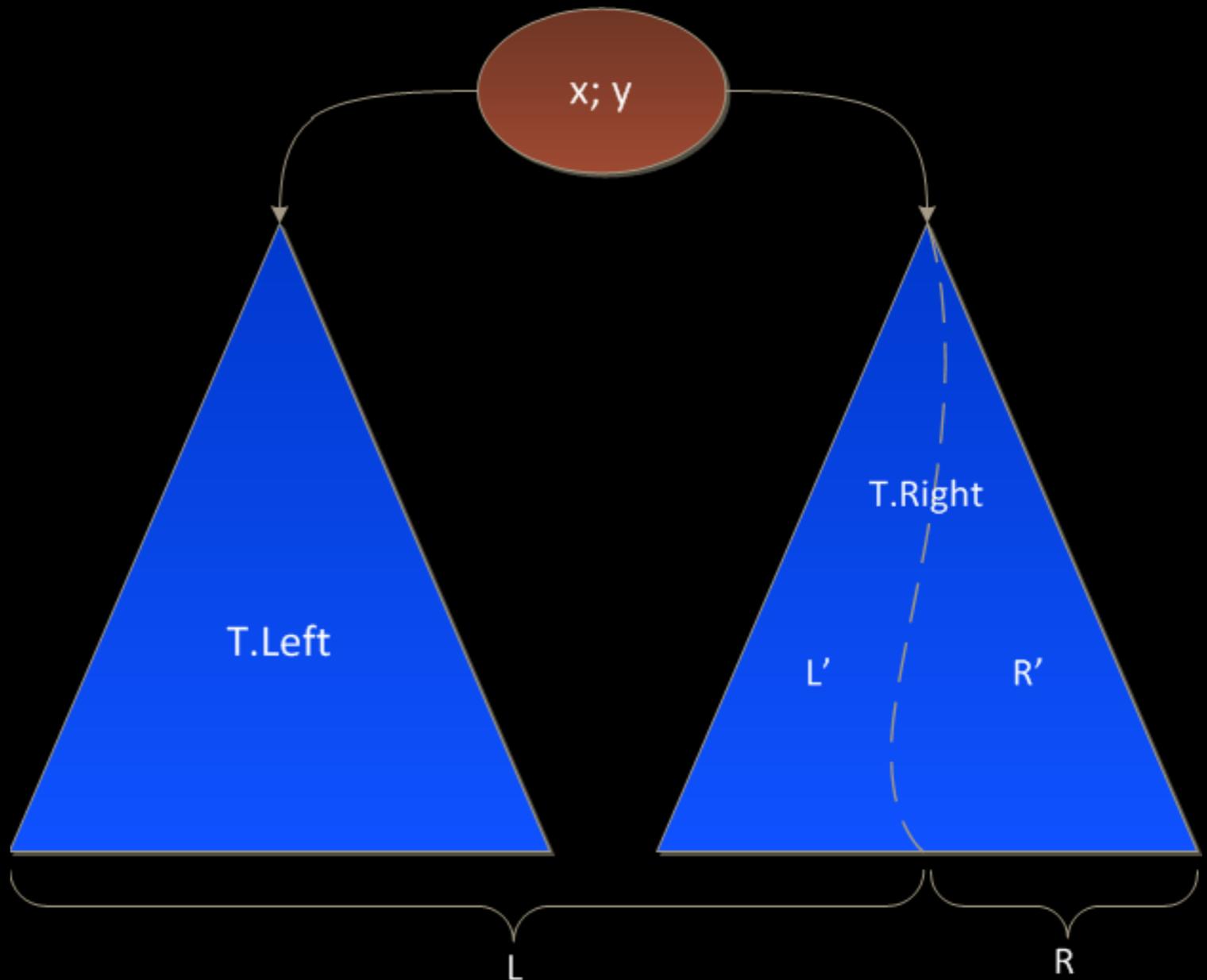
Задача 1

- Реализовать операцию Merge (все ключи одного дерева больше ключей другого)



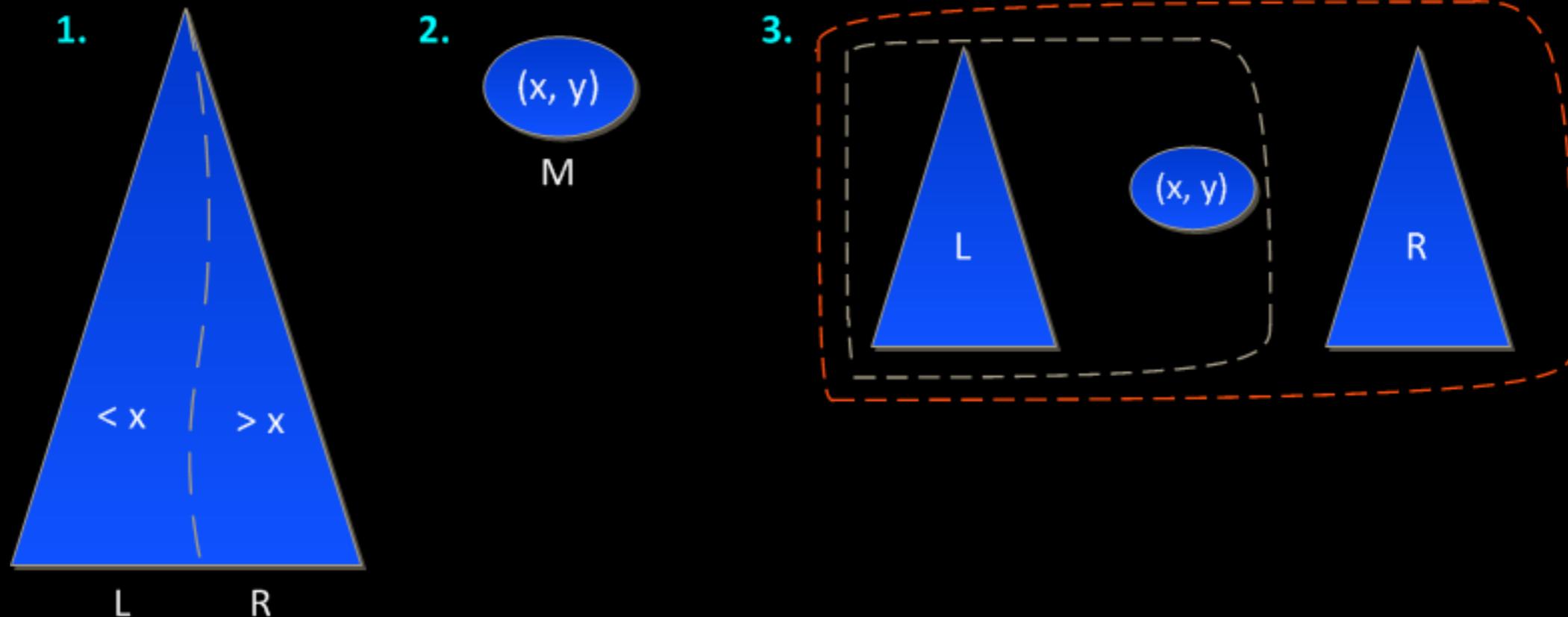
Задача 2

- Реализовать операцию Split по значению ключа x_0 .



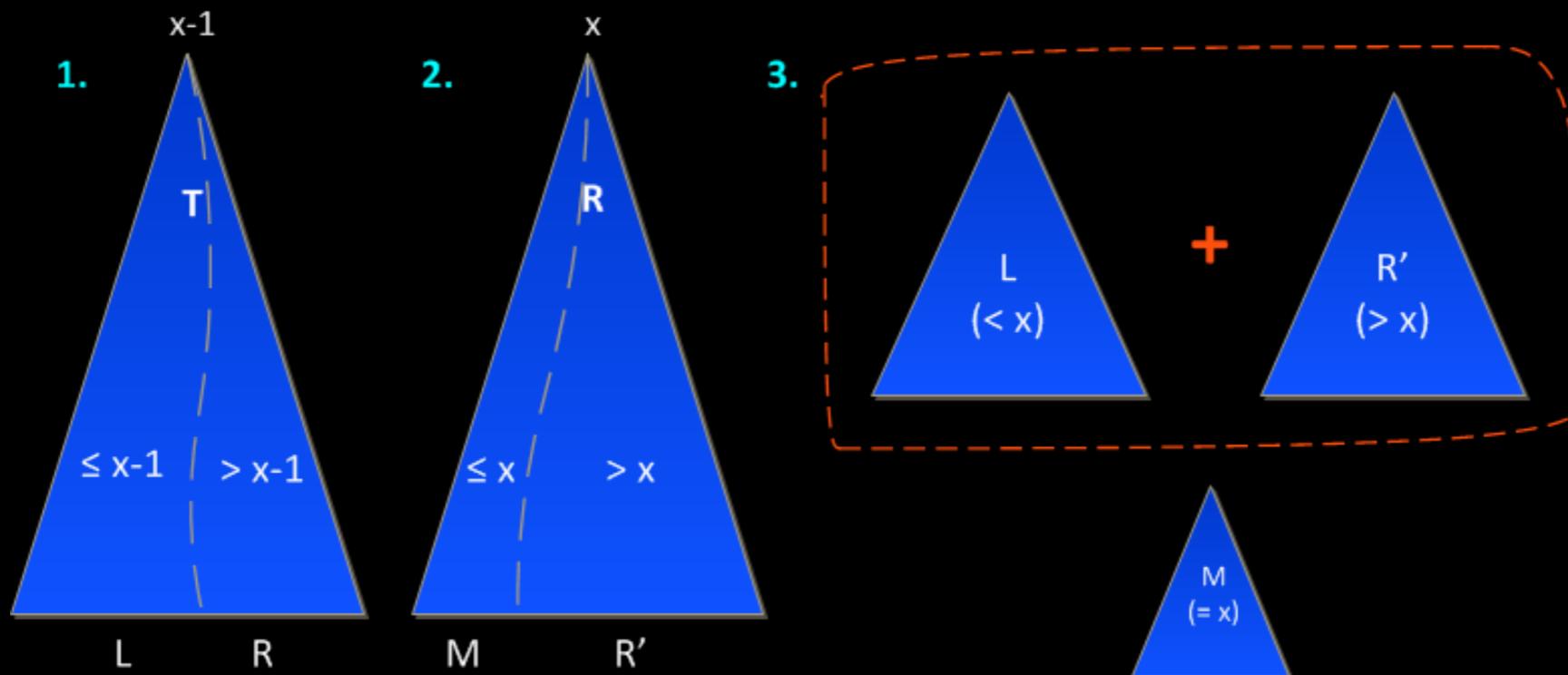
Вставка элемента

- Вставка элемента = разделение исходного дерева по ключу x и слияние одного из них с тривиальным (x, y) - y - случайный приоритет;



Удаление элемента

- Пусть надо удалить элемент по ключу x
- Допустим равные ключу элементы могут храниться в левом поддереве
- Тогда, сначала разделим дерево по ключу $x-1$, потом правый результат разделим по ключу x
- Среднее дерево и есть искомым результатом,
- Осталось объединить правое и левое деревья.



Резюме

- Дерамида имеет логарифмическую высоту => За логарифмическое время находит любой ключ
- Быстрые операции удаления и добавления элемента.

ДЗ

- Доделать операции с деоамидой и реализовать эффективный алгоритм построения дерамиды при ключах поступающих в порядке возрастания (см. <https://habrahabr.ru/post/101818/>)