



Treap

Pittsford Sutherland Programming Club

By Yizuo Chen, Yiyu Chen

Treap is another “Balanced binary search tree” – simplest one to write

Binary Search
Tree

Binary Heap

Treap

```
graph LR; BST([Binary Search Tree]) --- Treap[Treap]; BH([Binary Heap]) --- Treap;
```

The diagram illustrates the relationship between different data structures. On the left, there are two dark gray ovals: the top one contains the text 'Binary Search Tree' and the bottom one contains 'Binary Heap'. On the right, there is a dark gray rectangle containing the text 'Treap'. Two thin black lines connect the right side of each oval to the left side of the rectangle, indicating that a Treap is a combination of a Binary Search Tree and a Binary Heap. In the background, there are several curved, light blue lines on the left side of the slide.



Treap

- Use Binary search tree's properties to make it easy to find certain elements.
- Use Heap's properties to make it "balance".



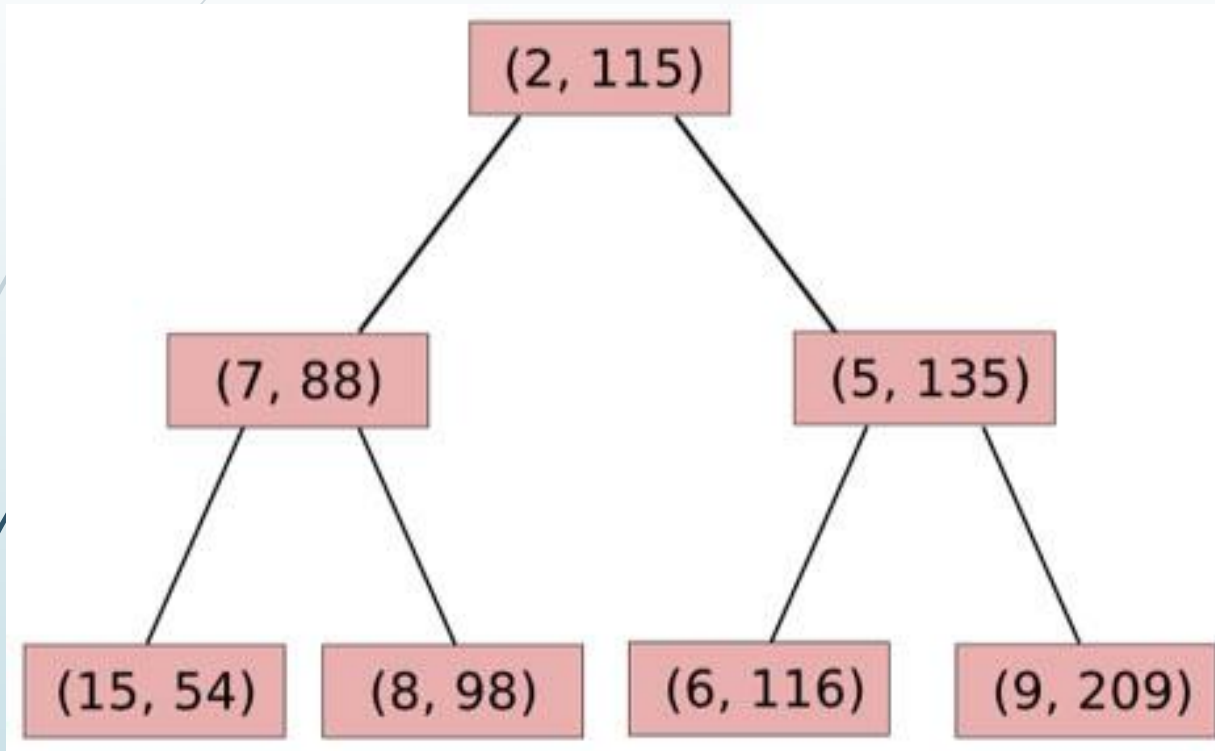
How?



Inside the Treap

- Besides left child, right child, value, nodes in Treap have another elements called **Priority value**.
- The priority value for each node is given by a **random number**; however, these priority values must satisfy heap order.

Example:



In each node, the first value is the priority value, and the second value is the element value.

As you can see, Treap is a binary search tree + Min Heap



How to keep Treap “balanced”?

➡ Use rotations!





Insertion

- The same way Binary Search Tree Insertion to find the position to insert.
- When find the position to insert the node, compare the nodes' priorities with their children's. Use Rotation to keep parents' priorities less than their children's priorities.



Deletion

- The same way Binary Search Tree deletion to find the position of the node to remove.
- Compare the priorities of two children, use rotation to make the child with smaller priority to be the parent. (In order to satisfy the heap property)
- Remove the node from Treap.



Time Complexity: $O(\log n)$