

# Priority Queue

Mar 1st, 2020

# Priority Queue

- An extension of a Queue Data Structure
- Properties:
  - Every Item in the Queue has a priority key attached to the item
  - The item with the high priority is deque before the items with low priorities.
  - If two items have the same priority, the deque will be determined by their order in the queue.

# What Priority Queue are for

- Scheduling
- Graph/Path Algorithms
- Queue Applications

# Central Priority Queue Operations (as an unsorted array)

- `Insert(item, priority)`: Inserts an Item with an priority -  $O(1)$
- `Get_First(self)` : Return the item with the highest priority -  $O(N)$
- `Remove_First(self)`: Delete the item with the highest priority -  $O(N)$
- `Is_Empty(self)`: Return if queue has no value -  $O(1)$
- `len(self)`: Return the length of the queue -  $O(1)$

# Priority Queue as a Heap

What is a heap:

- A Data Structure based on Complete Binary Tree
- Every node with the same depth in the heap should be filled before traversing down to another depth
- Min-Heap: All parents should be smaller than their descendants
- Max-heap: All parents should be bigger than their descendants

# Priority Queue as a Heap

How can you use heap as implementation for Priority Queue?

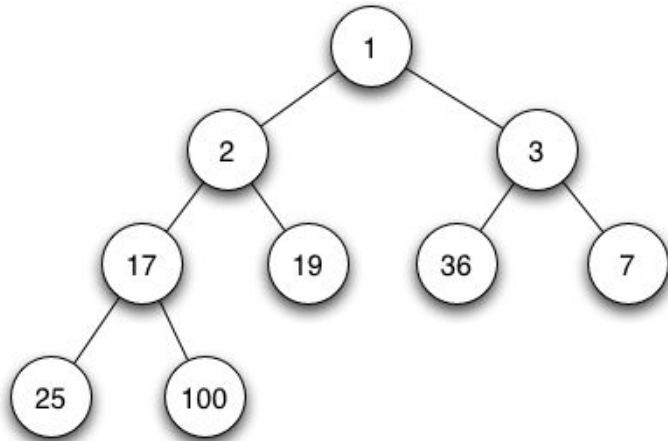
=> Create a hierarchy using the heap where the root node is the highest priority item, and the closer to root, the more high priority it is.

This way, operation such as `delete_first` and `get_first` will be done much easier, since we just have to pop/peek at the root node of the heap.

# Priority Queue (Insertion for Heap)

Rule to remember:

- Fill all the children of a depth from left to right before moving on to another.
- All parents must be smaller than the children



# Priority Queue (Insertion for Heap)

When insert a item into the queue:

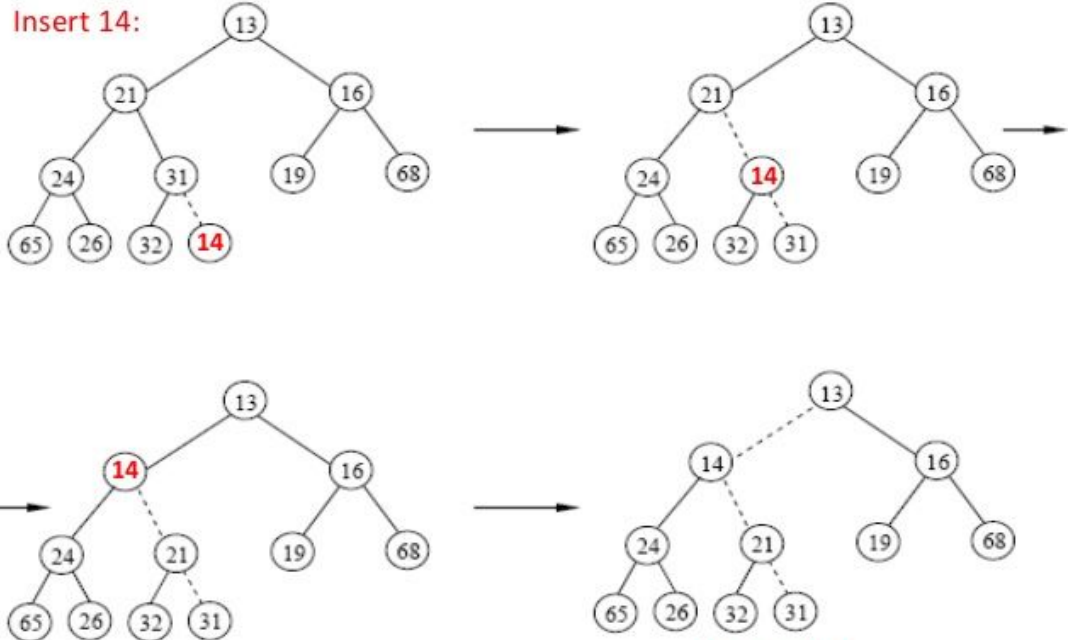
- Move to the available position and add the item as a leaf node.
- Case 1: If the item's priority is bigger than the parent => Done
- Case 2: If the item's priority is smaller than the parent:
  - Swap value of the node with its parent
  - Recursively check the priority of the parent with the ancestors until Case 1 is reached or the item becomes the root node
- What's the time complexity?  $O(\log N)$



# Priority Queue (Insertion for Heap)

Example of Priority Queue  
(as a Min Heap)

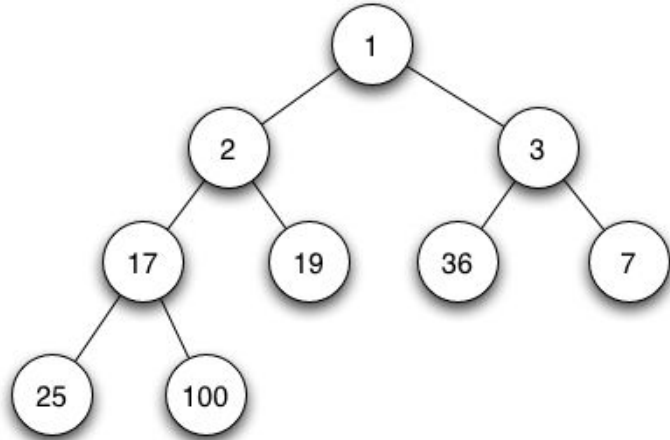
## Insert new element into min-heap



The new element is put to the last position, and **ReheapUp** is called for that position.

# Try one!

Insert 4 into the heap



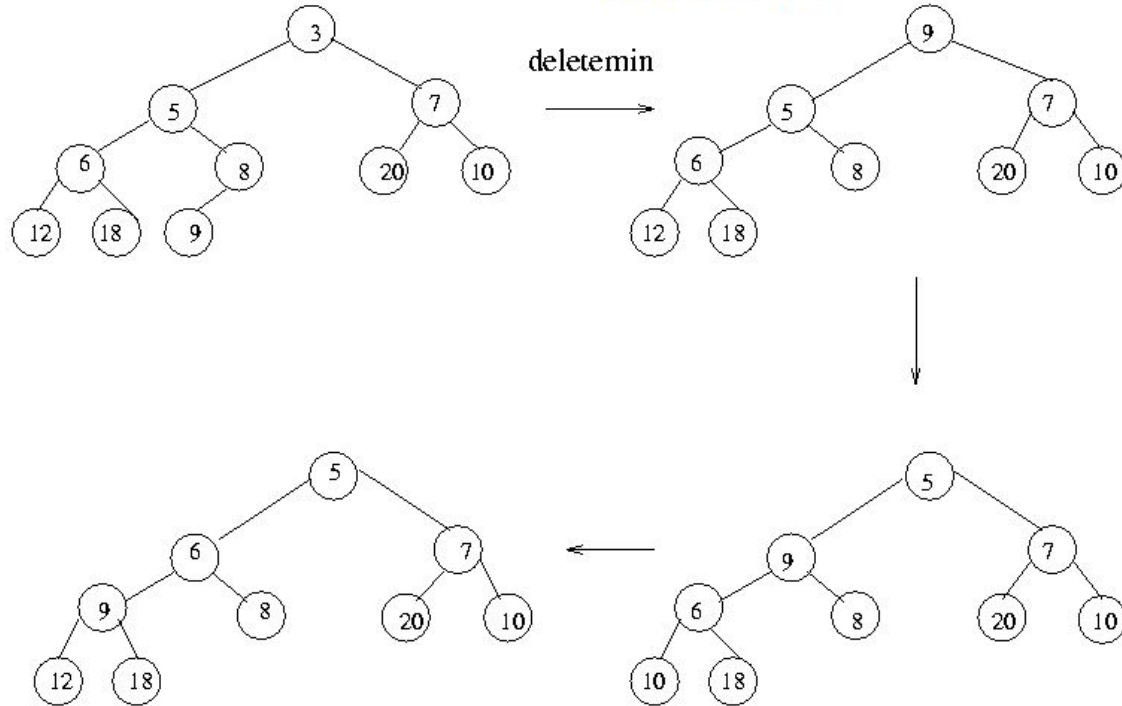
# Priority Queue (Deletion for Min-Heap)

Rule to remember:

- If  $\text{len}(\text{queue}) = 1$ , delete the root and return None
- Delete the root node and replace it with the last item in the heap
- Check if the new root is smaller than any of its children:
  - Case 1: If there's no children or the root priority is smaller than its possible children, then stay the same
  - Case 2: If there exist one child that is smaller than the root: Swap with the child and do the checking step with the new position of the root.
  - Case 3: If there exist two child that is smaller than the root: Swap with the left child and do the checking step with the new position of the root.
- Time Complexity:  $O(\log N)$

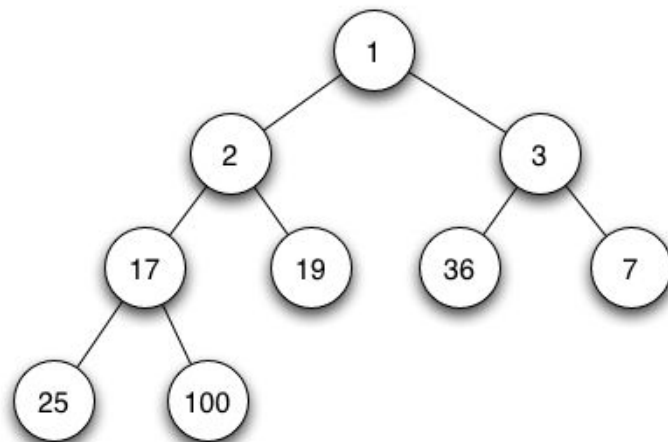
# Priority Queue (Deletion for Min-Heap)

Figure 6.3: Deletemin



# Try One!

Try delete



# Priority Queue (get\_first, len, is\_empty)

- get\_first: Return the value of the root =>  $O(1)$
- is\_empty: return root == None =>  $O(1)$
- len: Return a tracker that keep track of number of item inserted, deleted =>  $O(1)$

# Real World Example - EC Room Selection

- Purpose: Students with more credits will be prioritized to select the room first.
- Priority: Number of Credits, Item: Student
- Use Max-Heap as Priority Queue
- After insertion of all the students, with their given number of credits as priority, start delete\_first to get the student who get to choose the room first.

# Reference and Figure Source

- <https://visualgo.net/en/heap>
- <http://lcm.csa.iisc.ernet.in/dsa/node138.html>
- <https://www.geeksforgeeks.org/priority-queue-set-1-introduction/>
- Data Structure and Algorithm in Python