CSCI-SHU 210 Data Structures

Recitation10 Worksheet Heaps, and Priority Queue

Important nodes for this week's recitation:

- What is Priority Queue ADT
  - Store a collection of items, allow user to get min, or max.

- What is a heap.
  - [✓ Key(node) >= Key(parent(node)) or,]
  - Key(node) <= Key(parent(node))
  - This property is different from Binary Search Trees!!!

- How to store binary heap, or binary trees in an array.
  - Node index = i
  - Left child is at 2i + 1
  - Right child is at 2i + 2
  - Parent is at (i – 1) // 2

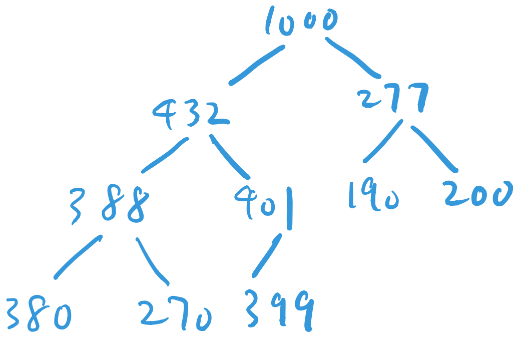
Part 1: binary-heaps

1. Consider the following array representation of a max-binary heap:

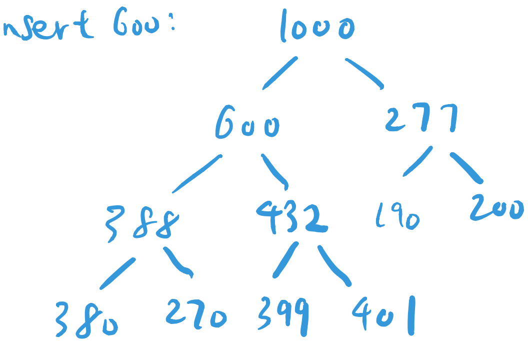
```
[1000, 432, 277, 388, 401, 190, 200, 380, 270, 399]
```

A. Show the tree representation of this binary heap.

B. Insert 600 into this binary heap. Show both the tree representation and the array representation after 600 has been inserted.



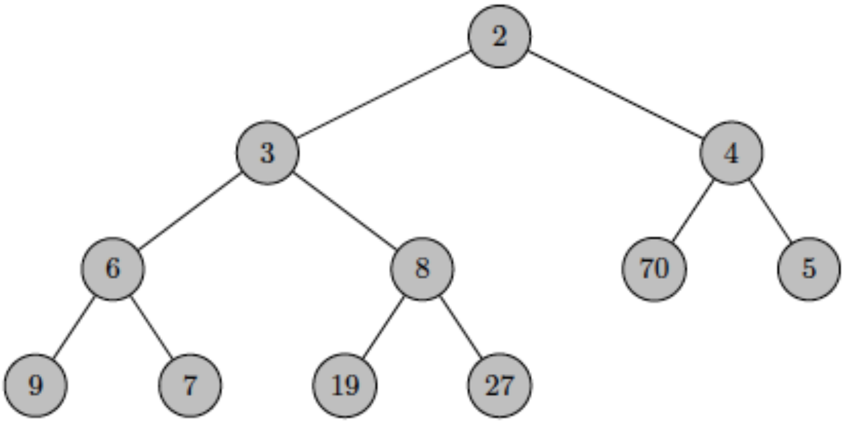
insert 600:



array:

[1000, 600, 277, 388, 432, 190, 200, 380, 270, 399, 401]

2. Consider the following tree representation of a min-binary heap:



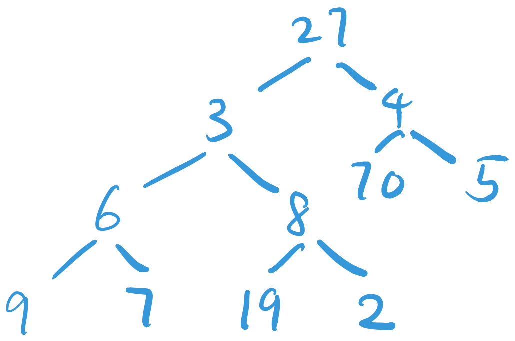
A. Show the array representation.
B. Show what happens when the root is removed by giving the tree representation of this binary heap.

2.A: [2,3,4, 6,8,70, 5,9,7,19,27]

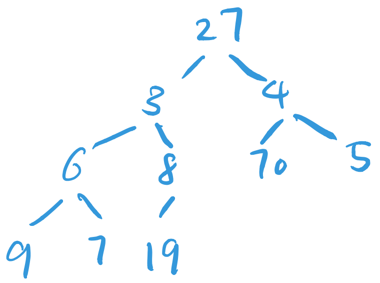
Your task 3: heap_priority_queue.py is an implementation for min_heap.

Now, modify the code, so the heap becomes a max_heap.

2.B: step1: exchange the root with the last node (②, ㉗.)

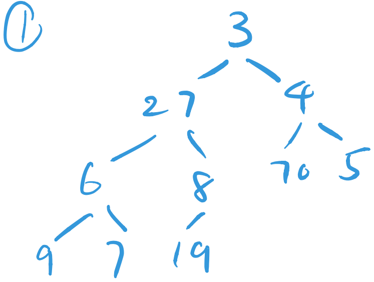


step 2: remove ② , which now at the last node .



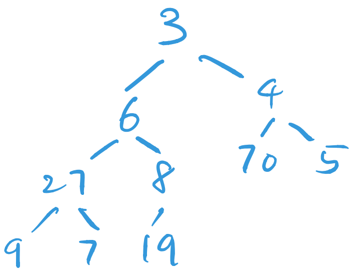
step3: maintain the heap's order;
do downheap operation

①



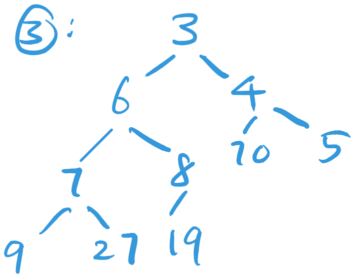
=>

②



<=

③:



Finished.