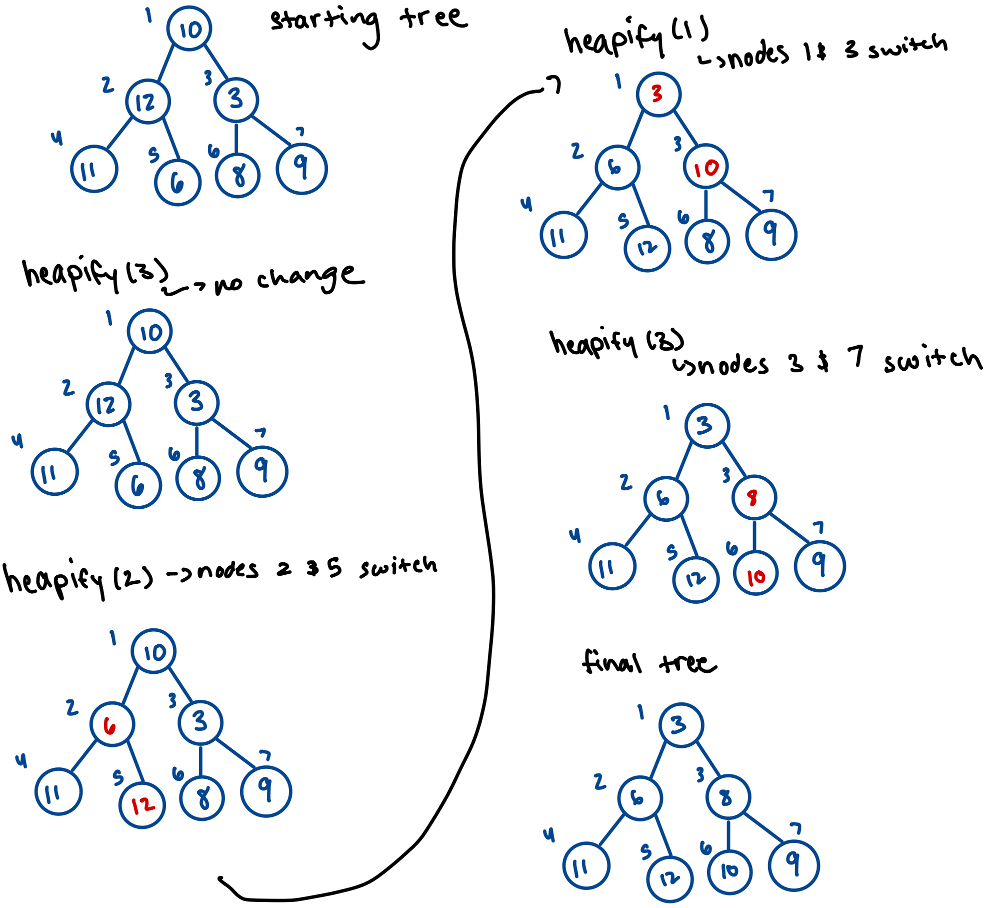
Part 4(a)

Perform the `buildHeap` (aka `makeHeap`) algorithm on the following array to create a **min-Heap** from the arbitrary array shown below. Show the state of the array as a binary tree after each iteration (call to `heapify()`) of the algorithm. (If that does not make sense review the lecture materials to review the `buildHeap` algorithm.)

- [10, 12, 3, 11, 6, 8, 9]


starting tree

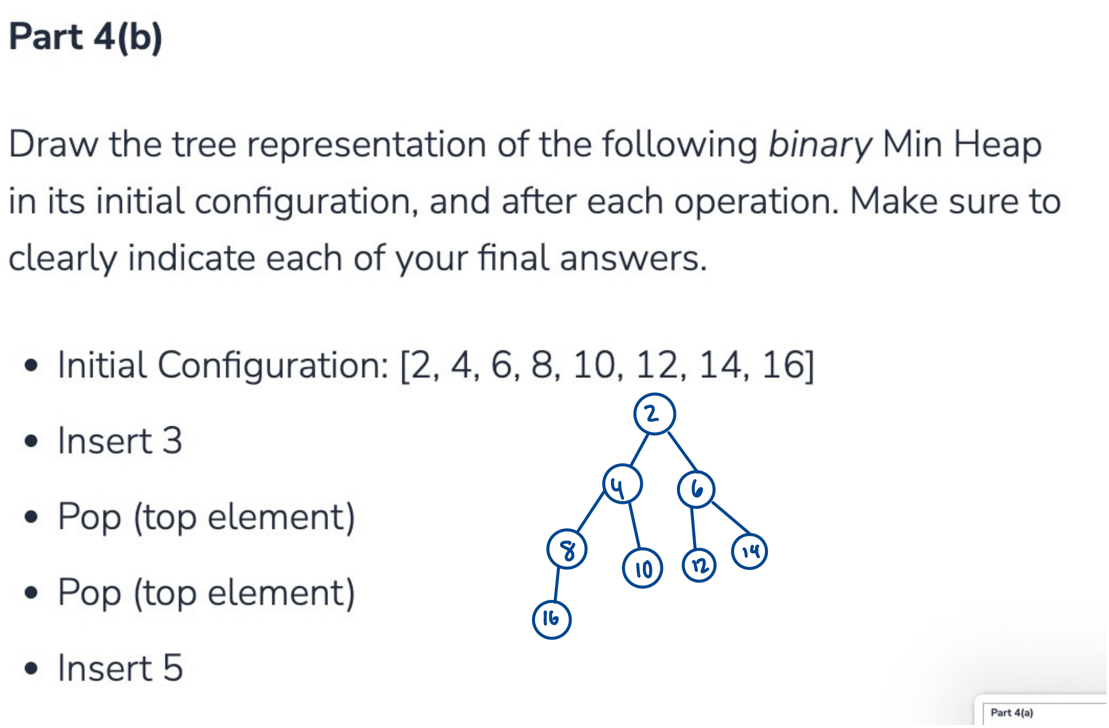
heapify (3) ↳ no change

heapify (2) -> nodes 2 & 5 switch

heapify (1)
↳nodes 1 & 3 switch

heapify (3)
↳nodes 3 & 7 switch

final tree

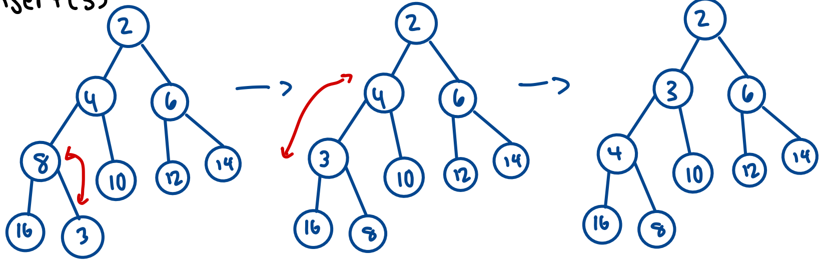


Part 4(b)

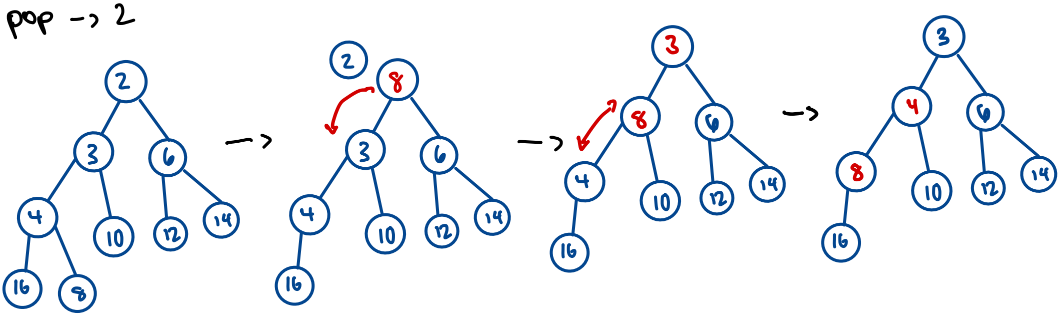
Draw the tree representation of the following *binary* Min Heap in its initial configuration, and after each operation. Make sure to clearly indicate each of your final answers.

- Initial Configuration: [2, 4, 6, 8, 10, 12, 14, 16]
- Insert 3
- Pop (top element)
- Pop (top element)
- Insert 5

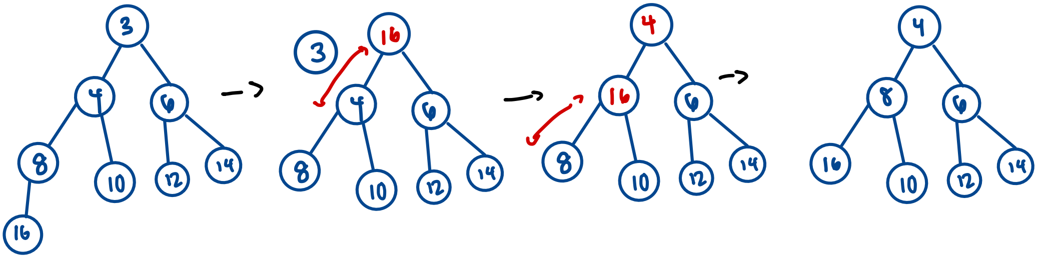
insert(3)



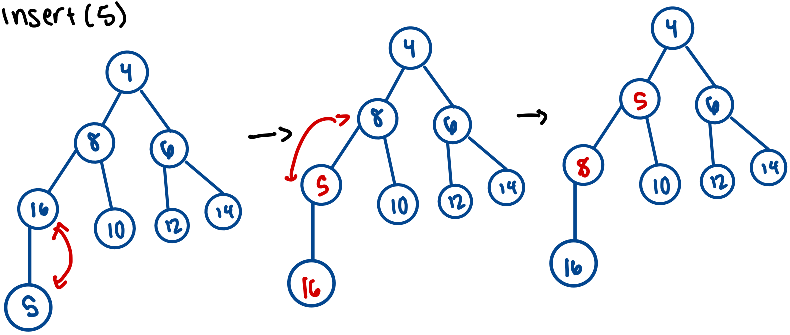
pop -> 2



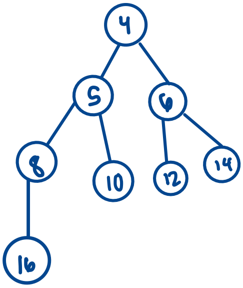
pop -> 3



insert (5)



final tree


4
5
6
8
10
12
14
16