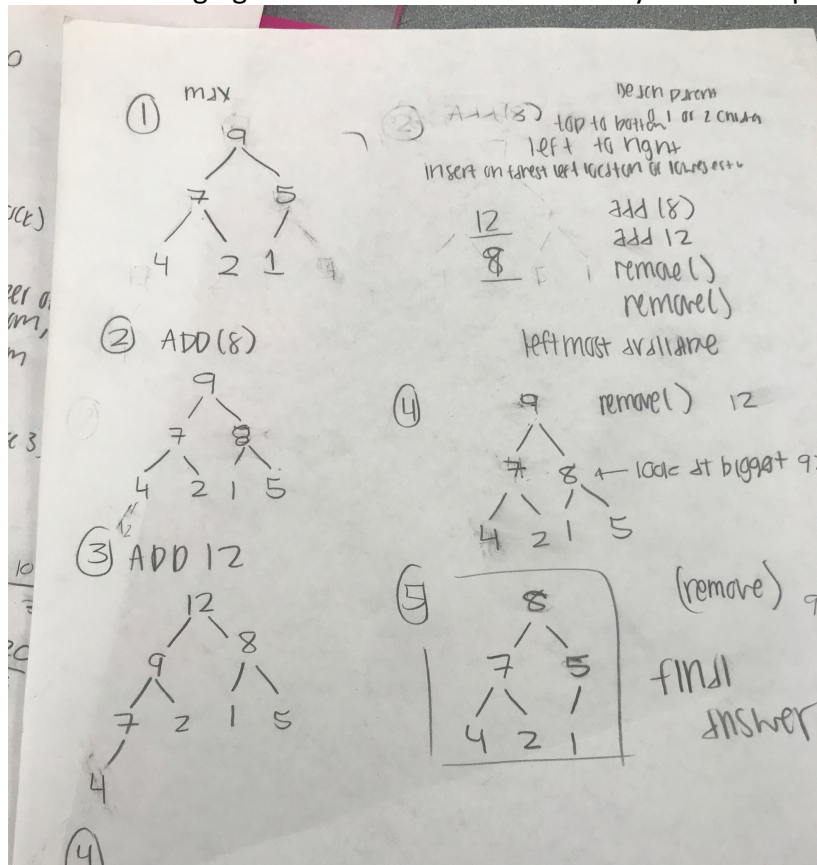


HW #5 (worked with Amara)

1. An algorithm that uses Priority Queue to sort a list of integers is Dijkstra's Algorithm. First you will mark the selected initial node with a current distance of 0 and the rest being null. Then set the non-visited node with the smallest current distance as the current node C. For each neighbor N, of the current node C add the current distance of C with the weight of the other node going down the path of C and N. If it is smaller than the current distance of N, set it as the new current distance of N. Mark the current node C as visited. If their non-visited nodes repeat.
2.
  - a.  $O(\log n)$  because it would have to look through each node in the unsorted array and rearrange them accordingly into a heap. Percolating up is  $\log n$ ,  $n$
  - b. Since a heap satisfies the heap invariant (min and max), the heap is already sorted therefore it would  $O(n)$  because you would just visit each node instead rearranging the elements. There is already a shortest path between two nodes.



- 3.
4. `public class GenericBox<T extends Number>`
- 5.

- a. When running the main method, it throws `IllegalArgumentException`, because the heap property is violated. The `compareTo` method in the main organizes the array based reverse alphabetical order however this

property is not being satisfied by the heap that is being passed through the main method.

b.

```
public int compareTo(StringMinHeapKey o) {  
  
    return -1*key.compareTo(o.key);  
}
```

c. Multiplying by negative one works because it allows us to reverse the priority of the tree.

8.

