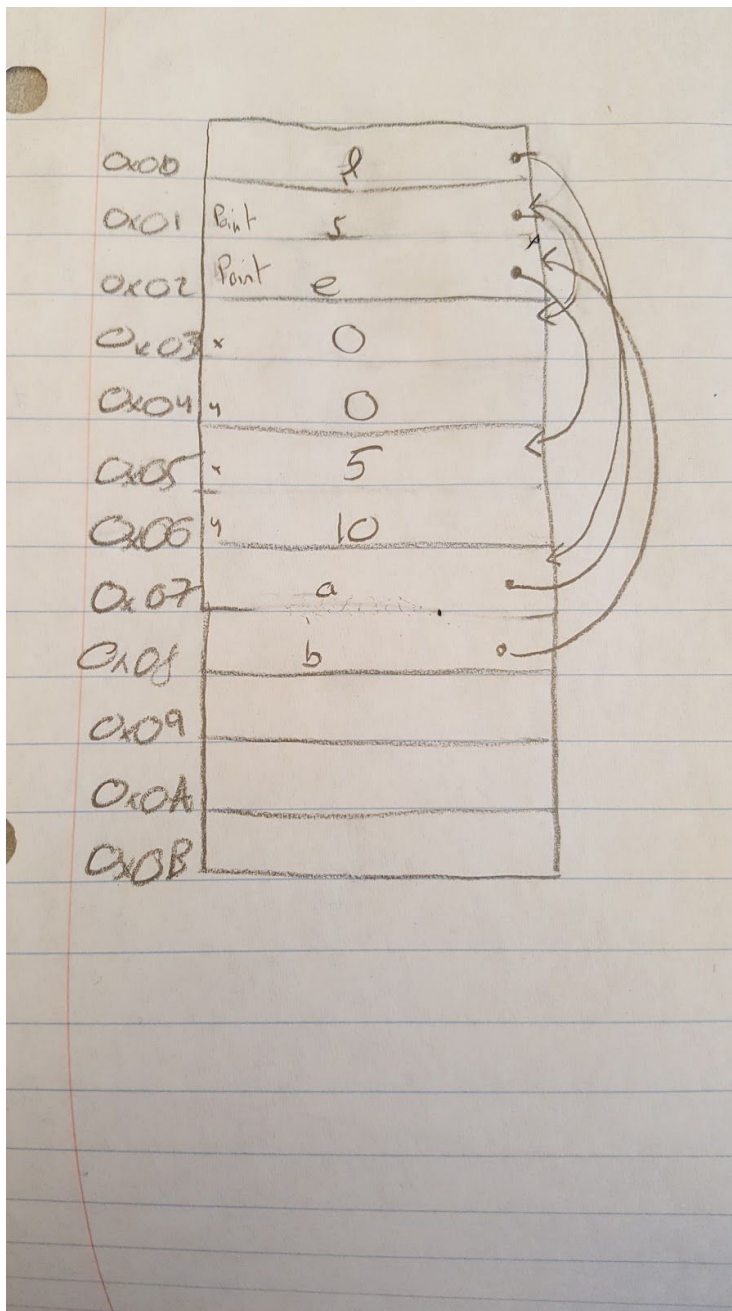


Aymeric Foyer

Question 1:



Question 2:

Problems are tasks given to us and algorithms are used to solve those problem. Programs are a way to get all the algorithms together to solve the problems given to us.

Question 3:

They can be different because  $1000 \cdot n^2$  and  $n \cdot 2$  are still the same time complexity but take very different times.

Question 4:

No. the "==" can also be used to check if two objects point to the same spot in memory whereas equals() only checks if their values are the same.

Question 5:

Pre order: 14,10,6,12,15,73,21

In order: 6,10,12,14,15,21,73

Post order: 6,12,10,21,73,15,14

The tree is neither full nor complete as 15 and 73 do not have 2 leaves (making the tree not full) and the leaves do not go left to right for each level (for example 73 and 21) making the tree not complete.

The tree does look like a binary search tree because the maximum amount of leaves a root currently has in this tree is 2 and minimum is 0.

Question 6:

DFS: A>G>D>H>I>E

BFS: G>D>B>E

Cycles: -B>C>J>I>D>B

-A>G>A

Question 7:

1. The list implementation won't work because the insert method enqueues to q and not the temp Queue. Also append does not increase the size of the queue.
2. Again, the append method does not increase the temp size and the insert problem I talked about above.
3. Fetch is a linear time complexity and a tight bound because the method copies the entire queue to find the particular value.
4. Remove is also a linear tight bound because you need to go through the entire queue and copy it to the temp to update it.
5. Insert is also linear and tight bound because like the two other methods, you need to go through the entire queue and store it in a temporary one.
6. A linked list or array would be better because an array can return the fetch as a constant and a linked list can do the insert and remove without creating a temporary list. The current implementation has neither of those.