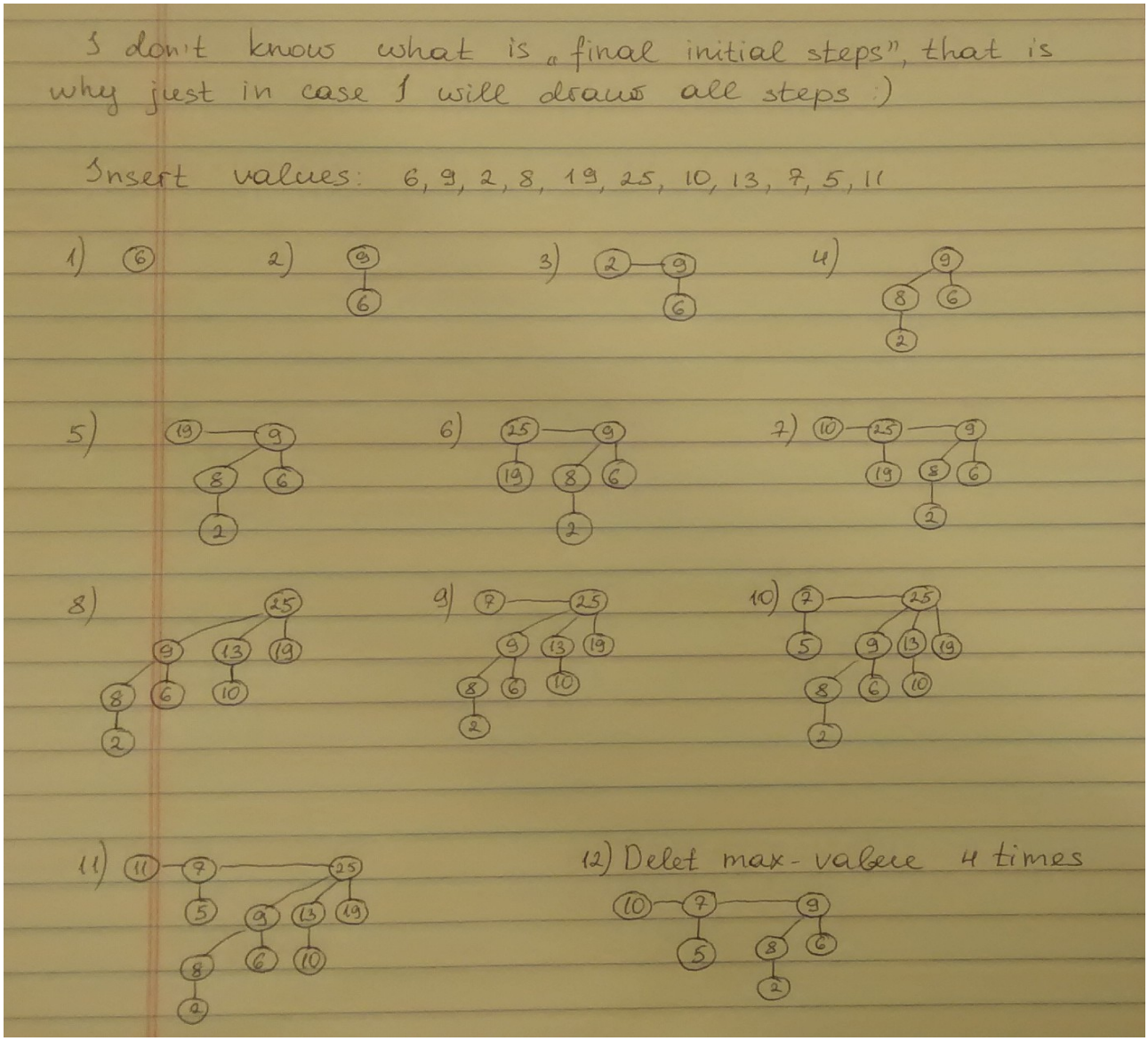
Homework #6

Union-find, Binomial Heap, Succinct trees

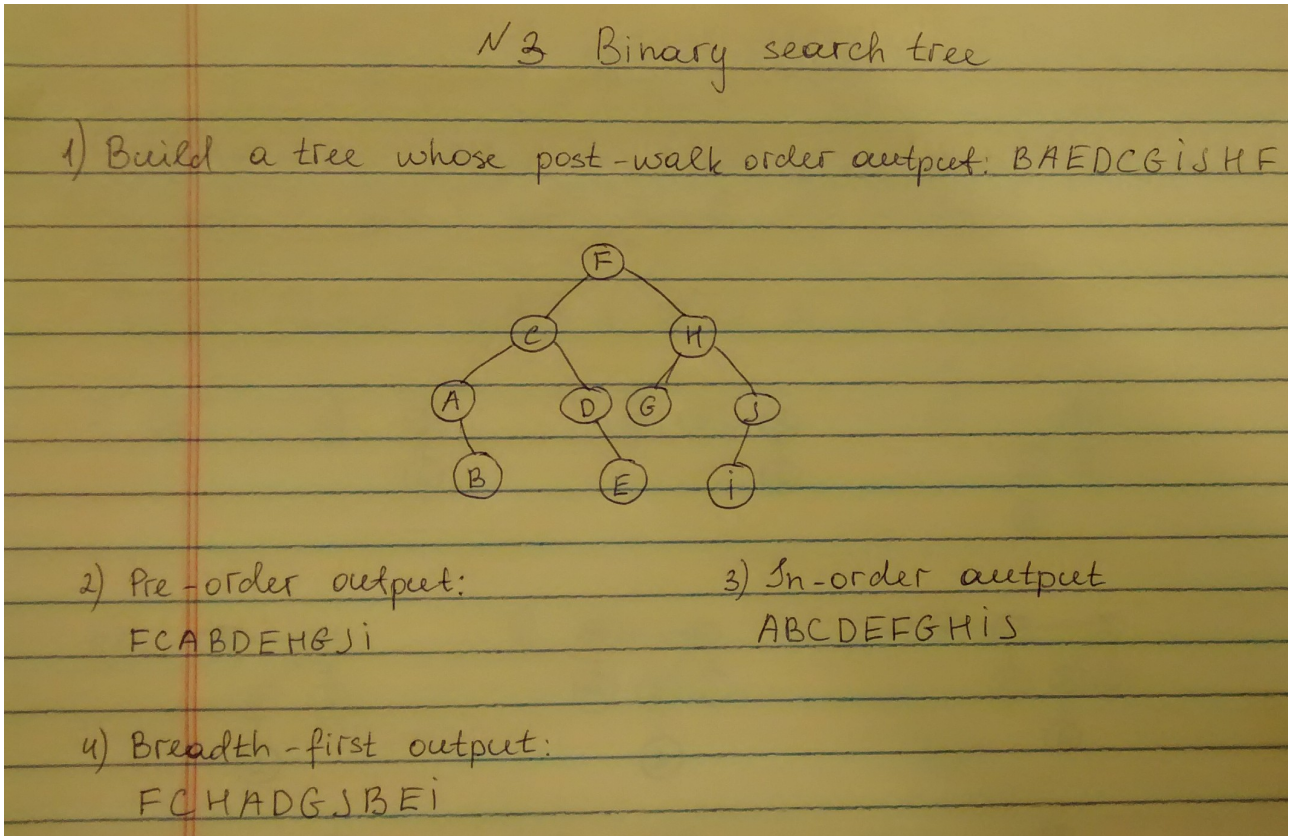
Anastasiia Konoplina

---

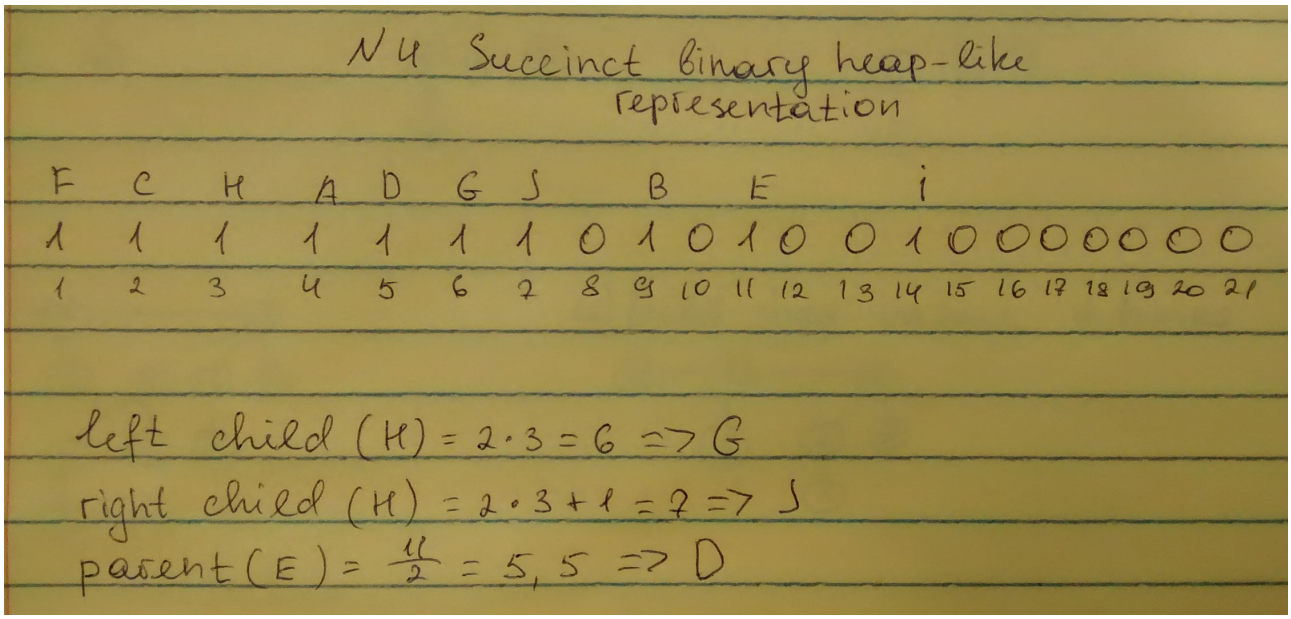
2. Insert values 6, 9, 2, 8, 19, 25, 10, 13, 7, 5, 11 into a max-priority Binomial heap. Draw the final initial steps and final state. Then delete the max value 4 times in a row. Draw the final state.



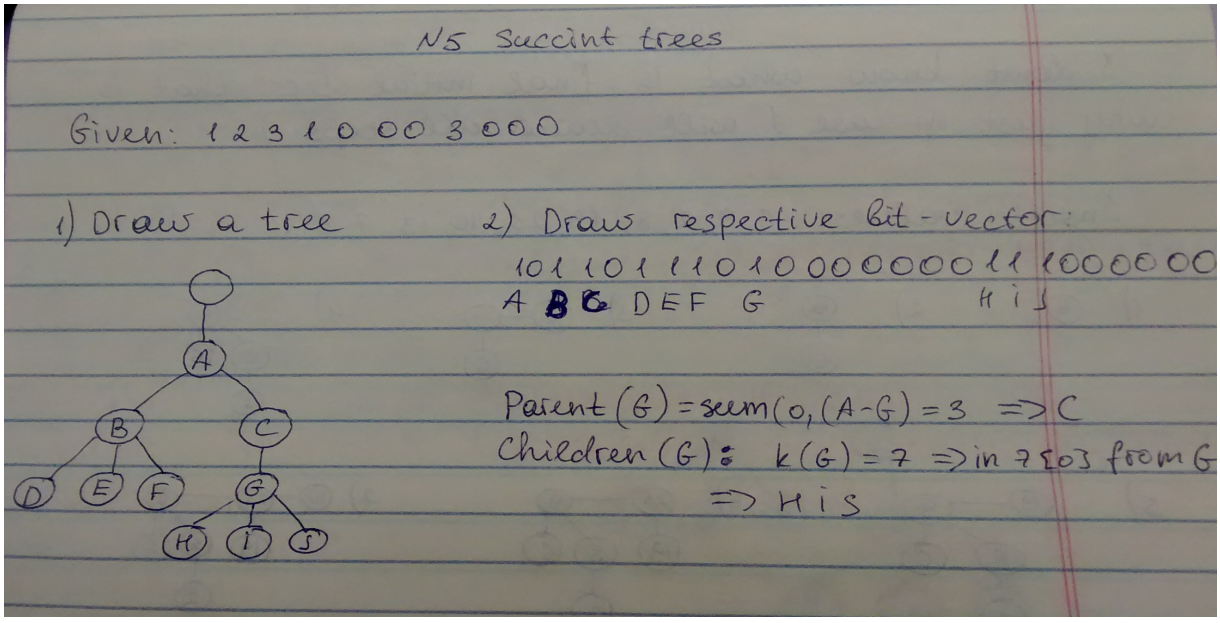
3. Draw a binary search tree whose values (characters) are in alphabetic order, and whose post-walk order output would yield: B A E D C G I J H F. What is it's pre-order and in-order output? What is the breadth-first order?



4. Represent the tree from Task 3 in a succinct binary heap-like manner with level-wise bit-vector representation, as outlined in the first example of the lecture slides. Show how to store keys (characters) and the respective tree. Show how to find the children of the node with H. And a parent of E.



5. A breadth-first node degrees of a generic tree (with a dummy root that does not hold information) are given as: 1 2 3 1 0 0 0 3 0 0 0. *Draw the tree, and the respective bit-vector for storing the above sequence. Label nodes depth-first A,B,C,D, ... Illustrate the operations for finding the parent and children of node G. Why does the parent finding work?*



Why does the parent finding work.

I think my theory is not correct, and it does not have any background, but anyways:

“0” separates nodes in each level, basically we can treat everything between separating zeros as one “logical node”.

If we did the same with binary tree every such “logical node” would contain 2 normal nodes. The formula of parent node in binary tree is x/2. But since every “logical node” contains 2 nodes, the formula is just x.

I know that it actually does not have sense, but this is the only idea I had.... sorry :(

---

6. (Bonus 2p) Write the same tree node degrees of Task 5 in depth-first order and in depth-first parenthesisation (0=open, 1=close). What are the (extra) necessary bit-vector operations to answer the questions about the size of the subtree, the parent, and the k'th child of a node?

Depth-first order: 12300013000, 110111000000010111000000

Depth-first parenthesis: ((()())((()()))) → 00010101100010101111