

## Trees & Search Trees Practice – 2 - Solution

### P1.

- a) In the tree below, the letters that you see in the nodes are NOT the actual VALUES of the nodes, but just LABELS so we can refer to these nodes. For example the root node is labeled with letter X, but the actual value of the node may be number 45, and the node labeled K may have value 7, and so on. There is no relation between the label given to a node and the actual hidden value of that node.

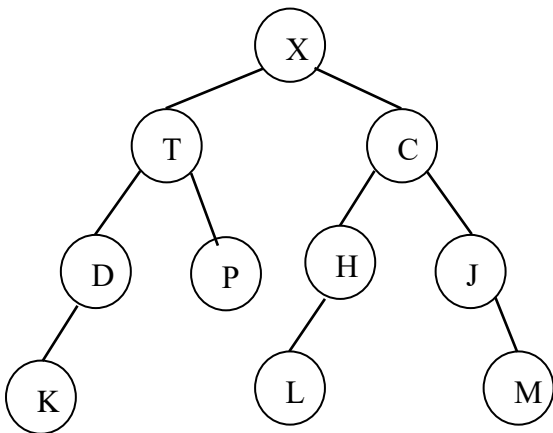
This question is about the order of the hidden VALUE, not the visible label of the nodes. This tree is a BST (binary search tree) according to the hidden value of the nodes.

The successor of node labeled K is: **\_\_D\_\_**

The successor of node labeled C is: **\_\_J\_\_**

The predecessor of node labeled X is: **\_\_P\_\_**

The predecessor of node labeled L is: **\_\_X\_\_**



### P2.

- a) When we split a 4-node of a 2-3-4 tree, which key (from that node) is being sent up to the parent?  
**A.** The left key   **B. The middle key**   **C.** The right key   **D.** There is no one single answer. We need to check the parent node to decide which key to push up.
- b) T / **F**   A 4-node in a 234 tree has 4 items inside of it. **It has 3 items**
- c) T / **F**   A 234 with N nodes has height  $O(N)$  if the worst case. E.g. when data is inserted in such a order that causes it to be unbalanced. **- 234 trees rebalance with each insertion. They cannot become unbalanced. Their height is at most  $\lg N$ .**