Seoul National Uiniversity

M1522.000900 Data Structure

Homework 5: Non-binary Trees (Chapter 6)

Computer Science & Engineering

2017-18538 Hwang Sun Young

Question1

E-K-M-P-S-J-B-F-H-G-C-Q-L-X-U-D-A

Question2

Table 1. Parent pointer

Index	0	1	2	3	4	5	6	7	8
Parent	11	-1	1	1	1	1	1	1	1

9	10	11	12	13	14	15
1	1	1	-1	1	1	-1

Question3

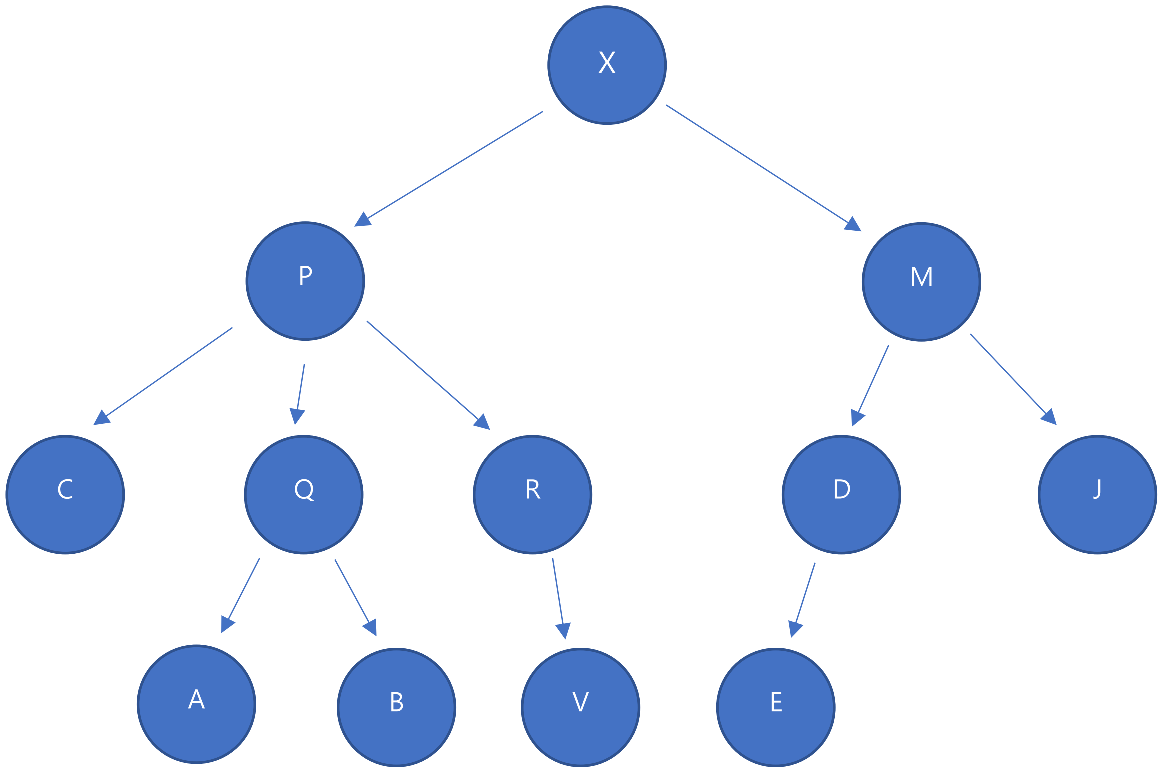
In "list of children" implementation, all nodes have a data value and a pointer to its list of children. And all children have a index and a pointer. The space overhead of the "list of children" implementation is

$$\frac{3P + I}{D + 3P + I}$$

In "left child/right sibling" implementation, all nodes have 3 pointers and a data value. The space overhead of the "left child/right sibling" implementation is

$$\frac{3P}{D + 3P}$$

Question4



Question5

-proof by mathematical induction

Base case: The number of leaves in a non-empty tree of 0 internal node is (K-1)0 + 1 = 1

Induction Hypothesis: Assume that the number of leaves in a non-empty full K-ary tree is (K-1)n+1, where n is the number of internal nodes.

Induction step: add K children to a leaf node in the tree with n internal nodes. Then the tree has one more internal node, and K-1 more leaf nodes. The tree has n+1 internal nodes. By Induction Hypothesis, we can assume that the number of leaves in a non-empty full K-ary tree is

(K-1)n + 1 + K-1, where n+1 is the number of internal nodes.

(K-1)n + 1 + K-1 = (K-1)(n+1) + 1

So the theorem still holds. Thus the theorem is correct by the mathematical induction.