

Homework 5

M1522.000900 Data Structure (2019 Fall)

2013-12815 Dongjoo Lee

Question 1.

- Post-order traversal: [E-K-M-P-S-J-B-F-H-G-C-Q-L-X-U-D-A].

Question 2.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Parent	1	x	0	1	3	1	1	1	2	3	1	0	x	1	1	x

Question 3.

Assume that there's n nodes. And also set the number of spaces of the memory of X to be $S(X)$. Regardless of the type of implementation, each node has a space for the data and the space overhead cause by for the implementation.

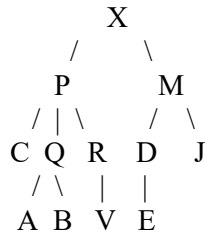
1) List of children implementation

- In array that contains every node, at each node, there is space for the data and for the pointer to point its parent and for the pointer to point its child if node has it. And at last, there is space for the index.
- There's $(n-1)$ child nodes cause every node has its parent except the root node.
- In array of child, at each of the child node, there is space for the pointer to point its sibling. And there is space for the index.
- To sum up, total data = $n(D+I+P)+(n-1)(P+I)$
So, as n is big enough, Data overhead / Total data = $(2I+2P)/(D+2I+2P)$
- When the operating system is by 32-bit CPU and data contains only integer value, we can assume that $D=P=I$, then, 4/5.

2) Left-child/right-sibling implementation

- In this implementation there is only one array that contains every node. At each node, there is space for the data and for the pointer to point its parent and for the pointer to point its leftmost child or right sibling. And at last, there is space for the index.
- The sum of the number of the leftmost child and right sibling is $(n-1)$.
- To sum up, total data = $n(D+P+(n-1)P+I)$.
So, as n is big enough, Data overhead / Total data = $(I+2P)/(D+I+2P)$
- When the operating system is by 32-bit CPU and data contains only integer value, we can assume that $D=P=I$, then, 3/4.

Question 4.



Question 5.

Let $S_k(x)$ be $(K-1)x+1$.

Proof.

The number of leaves in a non-empty full k-ary tree which has x internal nodes is $S_k(x)$

[Base case]

For $n=1$, full k-ary tree that has one internal node has exactly k children. So, $S_k(x)$ hold for $x=1$.

[Inductive Steps]

Assume that $S_k(x)$ holds for $x = n$, then, the number of leaves of full k-ary tree's number of leaves is $S_k(n)$.

When make tree to grow by $x=n+1$, there will be $n+1$ internal nodes.

And every internal nodes at $x=n$, need to be added K leaves.

Then the number of leaves is

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

$\therefore S_k(x)$ also holds for $x=(n+1)$.

QED.