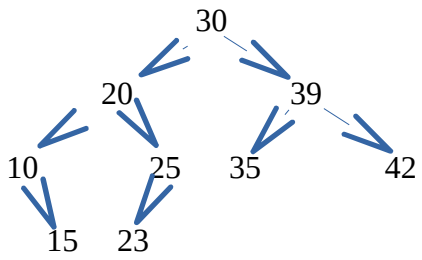
Ivan Alejandre
CST 370, ID: 8889
2/10/2020

Homework 5

1) Given the preorder sequence 30, 20, 10, 15, 25, 23, 39, 35, 42, find the postorder sequence

Binary tree:



Post order sequence: visit left subtree first, then right subtree, then root

Answer **D: 15, 10, 23, 25, 20, 35, 42, 39, 30.**

2a) inorder traversal: <left subtree> <root> <right subtree> order

10, 20, 30, 40, 50, 60, 70, 80, 90, 100

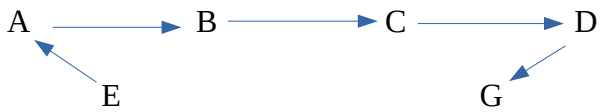
2b) preorder traversal: <root><left subtree><right subtree> order

80, 30, 20, 10, 40, 50, 60, 70, 90, 100

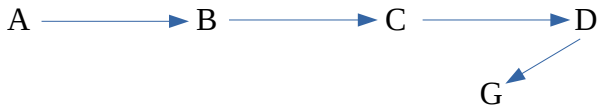
2c) postorder traversal: <left subtree><right subtree><root> order

10, 20, 40, 50, 70, 60, 30, 100, 90, 80

3) F goes first as it is the only source node. All other nodes have arrows going into and out of their nodes.



E is next source, remove.



The pattern should be apparent now, but the next source is A, followed by B, then C, then D, and finally G.

Topological sorting: **F, E, A, B, C, D, G**

4a) The problem is unclear whether the boys and boat start at the same side as the soldiers. For this answer, we'll assume that the boys are on the opposite side of the river. It'll take a total of 4 one way trips to ferry one soldier.

0) S_n	bbB	// initial set up
1) S_n + bB	b	// one boy takes boat across
2) S_{n-1} + b	SbB	// solider returns boat
3) S_{n-1} + bbB	S	// second boy takes boat back
4) S_{n-1}	SbbB	// both boys return boat to reset the loop

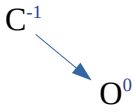
looking at the above, step 0 is our initial set up where the soldiers are on the left and the two boys (b) and the boat (B) are on the right.
In step 1, one boy takes the boat across the river.
Step 2, a soldier takes the boat back across to the other boy.
Step 3, the second boy on the right takes the boat back to the soldiers
Step 4, both boys return the boat to the right side. We're now back to our initial set up with the key difference of one soldier having crossed.

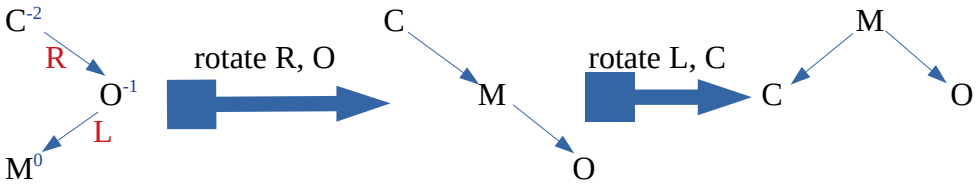
4b) If it takes 4 one way trips to ferry one soldier, then the boat will need to take **4*N one way trips** to ferry all soldiers. If we don't care that the boys end up on the same side of the river as they started, then it'll take 4*N-1 trips.

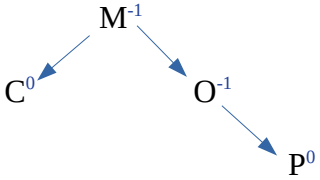
5a) No, node 6 has balance factor -2
5b) No, not a binary tree. Nodes 2 and 7 are in the wrong positions.
5c) Yes, nodes 50, 12, 23 have balance factor 1, all others have 0.

6) C, O, M, P, U, T, E, R --- rotate <direction>, <pivot node>

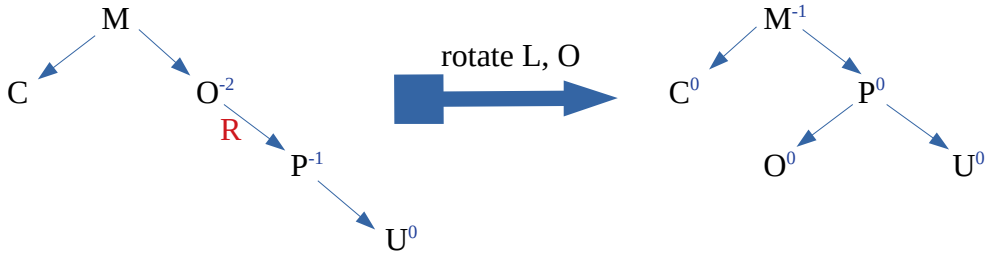
step 1: C

step 2: 

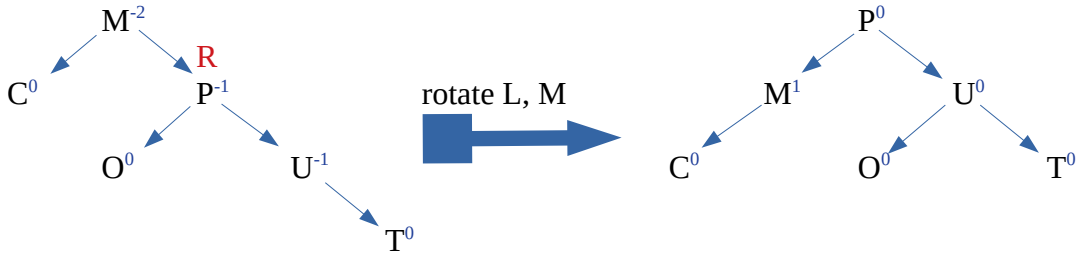
step 3: 

step 4: 

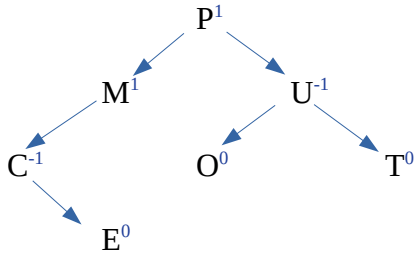
step 5:



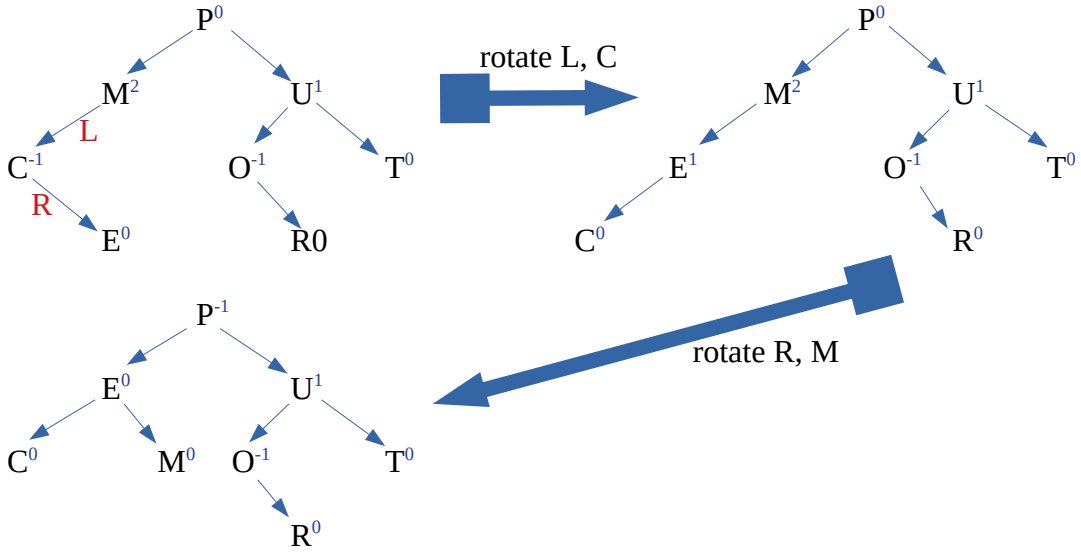
step 6:



step 7:



step 8:



7) I would select *insertion sort* as the algorithm of choice. The reason being is the inner while loop of insertion sort only executes once if the adjacent element is greater than the current. So, it behaves close to O(n). Since the input array is close to being sorted in non-descending order, time complexity is approximated to O(n) with the exception of the one out of order number.

8a) A[4] = [4, 3, 2, 1]
pairs: i,j → a[i], a[j]
0,1 → 4, 3
0,2 → 4, 2
0,3 → 4, 1
1,2 → 3, 2
1,3 → 3, 1
2,3 → 2, 1
6 inversions

8b) A[6] = [6, 5, 4, 3, 2, 1]
0,1 → 6, 5
0,2 → 6, 4
0,3 → 6, 3
0,4 → 6, 2
0,5 → 6, 1
1,2 → 5, 4
1,3 → 5, 3
1,4 → 5, 2
1,5 → 5, 1
2,3 → 4, 3
2,4 → 4, 2
2,5 → 4, 1
3,4 → 3, 2
3,5 → 3, 1
4,5 → 2, 1
15 inversions

8c) the largest amount of inversions is the sum of n from 0 to n -1

$$\sum_0^{n-1} \mathbf{n}$$