Object Oriented Analysis and Design: Assignment 0

Total Marks : 20

June 6, 2022

Question 1

Pre order and inorder traversals of a binary tree are `1, 21, 4, 15, 13, 16` and `4, 21, 15, 1, 13, 16` respectively. What is the post order traversal of the same binary tree?
Marks: 2 **MCQ**

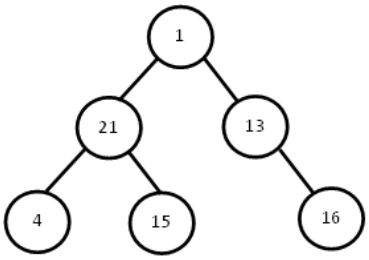
a) `16, 13, 15, 4, 21, 1`

b) `4, 15, 21, 13, 16, 1`

c) `4, 15, 21, 16, 13, 1`

d) `4, 21, 15, 16, 13, 1`

Answer: (c)
Explanation: The tree is as follows.



Hence, option (c) is correct.

Question 2

Suppose that you are given a binary search tree of integers and you are asked to find out the inorder successor of a given integer present in the tree. Which of the following is (are) true about the inorder successor of a given integer present in the tree?

Marks: 2 **MSQ**

a) The inorder successor of the given integer may be the integer present in its right child.

b) If the given integer is a leaf node, its parent must be its inorder successor.

c) If the given integer is in the root of the tree, it must not have an inorder successor.

d) If the right child of the given node is non-empty, the following code will return the inorder successor.

```
node* f (node *givenNode)
{
        node *p;
        p = rightChild of the givenNode;
        while (leftChild of p != NULL)
           p = leftChild of p;
        return p;
}
```

Answer: (a), (d)
Explanation: According to the definition of inorder successor, options (a) and (d) are correct.

Question 3

Suppose, you are searching for 37 in a binary search tree. Which of the following CANNOT be a possible sequence of comparisons?

Marks: 2 **MCQ**

a) 23, 40, 30, 38, 36, 37

b) 23, 40, 30, 38, 22, 37

c) 23, 40, 30, 35, 36, 37

d) 23, 45, 30, 38, 36, 37

Answer: (b)
Explanation: 40 is in the right of 23. So the tree rooted at 40 must have all integers greater than 23. 22 is in the right sub-tree of 23 which is not possible. Hence, option (b) cannot form the sequence of comparison in a binary search tree.

Question 4

Suppose that you are given a square matrix consisting of only 0 and 1. You need to find out some integer i so that i-th row (except the element a[i][i]) has all 0's and i-th column (except the element at a[i][i]) has all 1's. How many such i may be possible?

Marks: 2 **MSQ**

a) 2

b) 1

c) 0

d) O(n)

Answer: (b), (c)
Explanation: If i-th row has all 0's except the diagonal element, the columns except the i-th column cannot have all 1's. So, options (b) and (c) are correct.

Question 5

Consider a queue Q1, initially containing the following characters, with 1 at the front and 5 at the rear, as shown below.

Q1 : 1; 6; 7; 5

What are the contents of the queue after the following operations are performed in the given order?

Marks: 2 **MCQ**

Q1.enqueue (6)
Q1.dequeue ()
Q1.enqueue (5)
Q1.dequeue ()
Q1.dequeue ()

a) 7; 5; 6

b) 5; 5; 6

c) 5; 6; 5

d) 1; 6; 7

Answer: (c)
Explanation: After Q1.enqueue (6) operation, Q1 becomes: 1; 6; 7; 5; 6
AfterQ1.dequeue () operation, Q1 becomes: 6; 7; 5; 6
AfterQ1.enqueue (5) operation, Q1 becomes: 6; 7; 5; 6; 5
AfterQ1.dequeue () operation, Q1 becomes: 7; 5; 6; 5
AfterQ1.dequeue () operation, Q1 becomes: 5; 6; 5
Hence, option (c) is correct.

Question 6

Consider a max-heap of some numbers. Which of the following is not true about a max-heap of numbers? *Marks: 2* **MCQ**

a) The second largest element is the number stored at the left or right child of the root node of the max-heap.

b) The smallest element is found only at the leaf nodes of the max-heap.

c) The number of swap operations to initialize a max-heap is O(nlogn).

d) The number of swap operations to find out the smallest number in a max-heap is O(logn).

Answer: (d)
Explanation: According to the properties of the max-heap.

Question 7

A queue implemented in terms of a linked list has to maintain two pointers one pointing to its *front* and another pointing to its *rear*. You want to use a single pointer variable to access both these pointers in constant time. Which of the following is true in this context?

Marks: 2 **MCQ**

a) Such a queue can be realized by a singly connected linked list.

b) Such a queue can be realized by a doubly connected linked list.

c) Such a queue can be realized by a singly connected circular list with the pointer variable pointing to its *front*.

d) Such a queue can be realized by a singly connected circular list with the pointer variable pointing to its *rear*.

Answer: (d)
Explanation: If a queue is realized by a singly connected circular list with the pointer variable pointing to its *rear* then the *front* can be accessed by going to the next node from the node accessed by the pointer variable in one step.
Hence, option (d) is correct.

Question 8

Suppose, you are given two sorted lists of size m and n. Each list contains non-duplicate integers. There is no common integer in these lists. $m < n$. How many comparisons are required to merge these two lists in the best case?

Marks: 2 **MCQ**

a) $m * n$

b) $m + n - 1$

c) m

d) n

Answer: (c)
Explanation: The best case arises if all elements in the smaller list is less than the smallest element in the larger list. In that case m number of comparisons are required.
Hence, option (c) is correct.

Question 9

Stack A has the entries 1, 2, 3 with 1 at the top. Stack B is empty. When an integer is popped out of stack A, it can either be printed immediately or pushed to stack B. An integer popped out of the stack B can only be printed. Which of the following permutations of 1, 2, 3 is not possible?

Marks: 2 **MCQ**

a) 1, 3, 2

b) 3, 1, 2

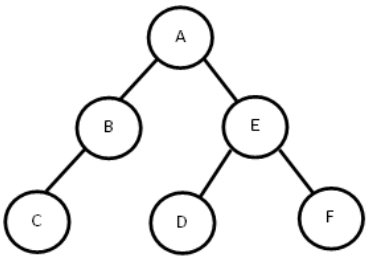
c) 2, 1, 3

d) 3, 2, 1

Answer: (b)
Explanation: Option (a): popA_and_Print(); popA_and_PushB(); popA_and_PushB(); popB(); popB()
Option(b): popA_and_PushB(); popA_and_PushB(); popA_and_Print(); Now stack B has 2, 1 with 2 at the top.
Option(c): popA_and_PushB(); popA_and_Print(); popB(); popA_and_Print()
Option(d): popA_and_PushB(); popA_and_PushB(); popA_and_Print(); popB(); popB()
Hence, option (b) is correct.

Question 10

Consider the following tree.



Consider the following algorithm.

Create an empty stack S and push root node to S.
while $S \neq$ empty **do**
  Pop an item q from S and print it.
  Add the left child of q to S, if available.
  Add the right child of q to S, if available.
end while

What will be printed if the above algorithm is executed on the given tree (from the root)?

Marks: 2 **MCQ**

a) A, E, F, D, B, C

b) A, B, C, E, D, F

c) A, B, C, D, E, F

d) C, B, A, D, E, F

Answer: (a)
Explanation: The tree is traversed as Data-Right-Left order.
Hence, option (a) is correct.