Final Exam

Name : Mosnoor Mofiz Arman

ID : 1921079642

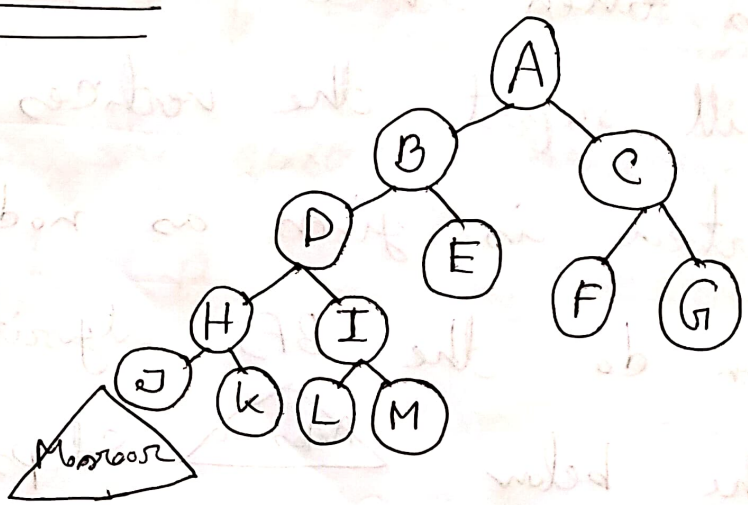
Course : CSE225

Section : 8

Ans to the QNO-03

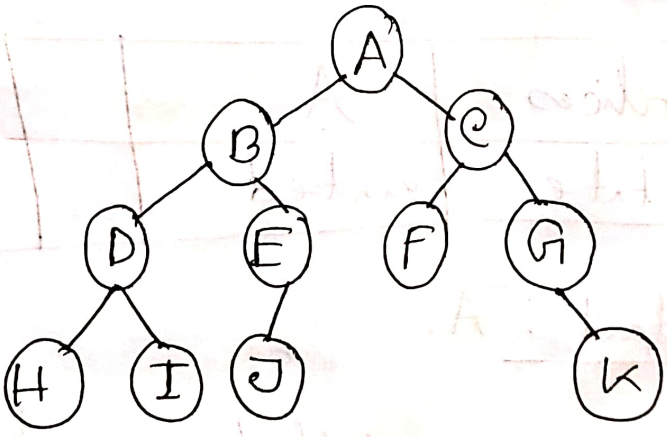
(i)

Both full and complete:



(ii)

Neither full nor complete:



Ans to the QNO : 01

For the BFS traversal, a queue Q is taken which is initially empty and will report the vertices. The source vertex is given as node A. So, we can do the BFS algorithm by following the below steps:

(1) Insert vertex A into the queue and mark it visited.

Vertices	A					
State	visited					

Output : A.

(2). The unvisited neighbours of vertex A in alphabetical order are vertices B, C and D.

Vertices	A	B	C	D		
State	Visited	Visited	Visited	Visited		

Output : A B C D.

(3). Remove the front element which is vertex A from the queue. The new front element is vertex B.

Vertices	B	C	D	E		
State	Visited	Visited	Visited	Visited		

Output : A B C D E

(4). Remove the front element which is vertex B from the queue. The new front element is vertex C. But vertex C has no unvisited neighbours. So, remove vertex C from the queue. The new front element is vertex D.

Vertices	D	E	F			
State	Visited	Visited	Visited.			

Output : A B C D E F.

(5). Remove the front element which is vertex D from the queue. The new front is vertex E. But vertex E has no unvisited neighbours. So, remove vertex E from the ~~queue~~ queue. The new front is vertex F. But vertex F also has no unvisited neighbours. therefore, remove vertex F from the queue.

So, the queue is empty and all ~~vertex~~ vertices have visited.

Thus, the BFS traversal is:

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$

(ABCDEF).

Now, for the DFS traversal, a stack S is taken which is initially empty and will reparted the vertices discovered and the state of the vertices. So, we can do the DFS algorithm which source vertex is given as node A by following the below steps:

(1). Push vertex A into the stack and mark it visited.

Vertices	A						
State	Visited.						

Output : A.

(2). Unvisited neighbour of vertex A in alphabetical order is vertex B.

Vertices	A	B				
State	Visited	Visited				

Output : A B

(3). Unvisited neighbour of vertex B in alphabetical order is vertex C.

Vertices	A	B	C			
State	Visited	Visited	Visited			

output : A B C.

(4). Vertex C has no unvisited neighbour. So, pop it from the stack and backtrack from vertex C to vertex B.

Vertices	A	B				
State	Visited	Visited				

output : A B C

(5). Unvisited neighbour of vertex B in alphabetical order is vertex E.

Vertices	A	B	E			
State	visited	visited	visited.			

output : A B C E

(6).

Vertices	A	B	E	D.		
State	visited	visited	visited	visited		

output : A B C E D.

(7).

Vertices	A	B	E	D	F	
State	visited	visited	visited	visited	visited.	

Output : A B C E D F.

(8).

Vertices	A	B	E	D.		
State	visited	visited	visited	visited		

Output : A B C E D F.

(9)

Vertices	A	B	E			
State	visited	visited	visited			

Output : A B C E D F.

(10)

Vertices	A	B				
State	visited	visited				

Output : A B C E D F.

(11).

Vertices	A					
State	visited					

Output : A B C E D F

(12)

Vertices						
State						

Output : A B C E D F

So, the stack is empty and all vertices have visited.

Therefore, the DFS traversal is:

A → B → C → E → D → F

(A B C E D F).

Ans to the QNO: 02

Given,

Tibonacci(n) = 2 * Tibonacci(n-1) + 3 * Tibonacci(n-3).

where base case:

Tibonacci(0) = 1, Tibonacci(1) = 1, Tibonacci(2) = 1;

Now,

```
Tibonacci(n)
if (n==0)
{
    return 1;
}
if (n==1)
{
    return 1;
}
if (n== 2)
{
  return 1;
}
else
{
  return (Tibonacci (n-1)+2 + 3* Tibonacci(n-3));
}
```

Ans to the QNo : 04

(a) Dequeue an item from Queue

$= O(n)$.

(b). Push an item into a stack

$= O(1)$

(c) Sorting N items using Merge

short $= O(n \log n)$.