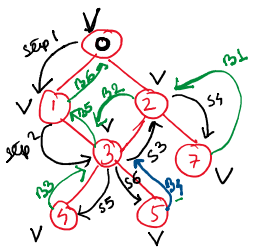
20 march Y section

20 March 2025 08:32



① Starting node :- 0th node

DFS traversal order -

0, 1, 3, 2, 7, 4, 5

Backtrack (BS)

Forward Step (S)

DFS Code → Recursion (System internally stack data structure).

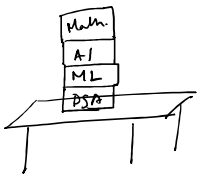
Adjacency list -

Stack - Feature - LIFO / FILO

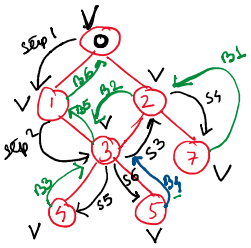
→ Insertion - Top

→ Deletion - Top

Stack of Books



Mathematics



1 Starting node :- node -

DFS traversal order -

0, 1, 3, 2, 7, 4, 5

Backtrack (BS)

Forward Step (S)

Ans list - 1 2 4 3 7 5 6

Adj list → List of List.

1 → 2 3

2 → 1 4

3 → 1 4 7

4 → 2 3 5 6

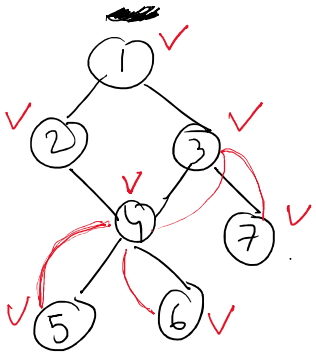
5 → 4

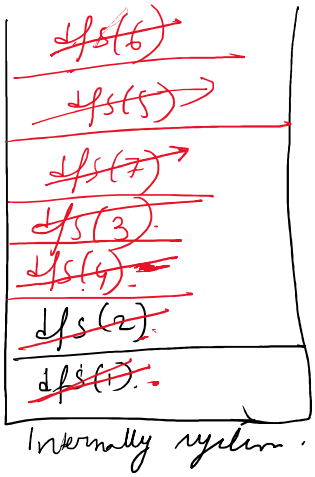
6 → 4

7 → 3

Pseudocode

```
dfs(Start_node)
{
  vis[start_node] = 1;
  list.add(start_node);
  for(Integer itr : adj.get(start_node)){
    if(visit[itr] == false){
      dfs(itr);
    }
  }
}
```





Internally system.

DFS start node 1.

visited array

0	1	2	3	4	5	6	7
0	1	0	1	1	0	1	0

100	25	6	7
0	1	2	3

for(i=0; i<arr.size; i++)

for(Integer itr : arr) {

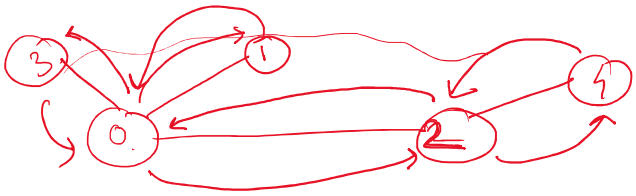
```
for(i=0; i<arr.size ; i++)
    print("arr [i]" )
```

100 25 6 7.

```
for( Integer itr : arr) {
    print(itr);
}
```

100 25 6 7

```
adj.get(0).add(2);
adj.get(2).add(0);
adj.get(0).add(1);
adj.get(1).add(0);
adj.get(0).add(3);
adj.get(3).add(0);
adj.get(2).add(4);
adj.get(4).add(2);
```



0 → 2 1 3

1 → 0

2 → 0 4

3 → 0

4 → 2

1. Fractional Knapsack
2. Activity Selection Problem.