① Starting node :- 0th node :-

DFS traversal order -
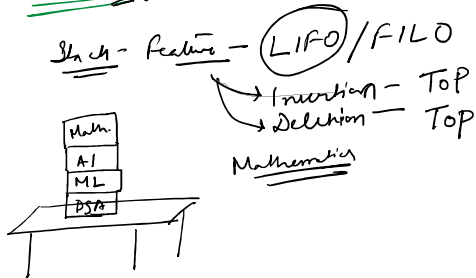
0, 1, 3, 2, 7, 4, 5

Backtrace (B)
Forward Step (S)

DFS code → Recursion (System internally stack data Structure).

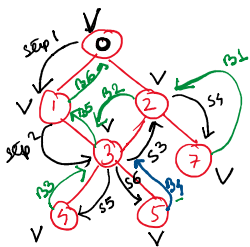Adjacency list -

Stack - Feature - (LIFO)/FILO

→ Insertion - TOP
→ Deletion - TOP

Mathematics

Stack of Books

| Math |
| A1 |
| ML |
| DSA |

1 Starting node :-   node -

DFS traversal order -

0, 1, 3, 2, 7, 4, 5
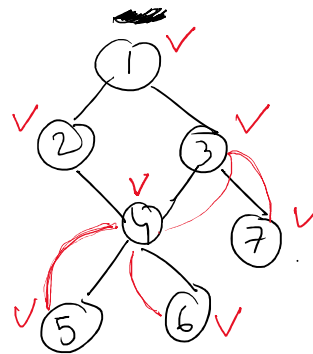
Backtrace (B)
Forward Step (S)

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);
    }
  }
}
3 3 ← 3

DFS Start   node 1-

dfs(6)
dfs(5)
dfs(7)
dfs(3)
dfs(4)
dfs(2)
dfs(1)

Internally system.

visited
array

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

| 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]")
```
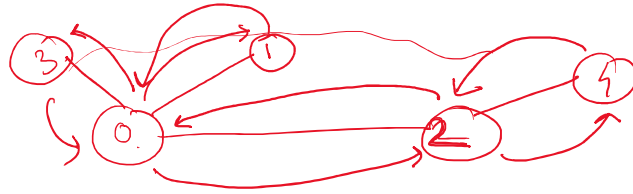
$$\boxed{100 \quad 25 \quad 6 \quad 7}.$$

```
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 KnapSach

② Activity Selection Problem.