

```

void Graph::DFS(){
    bool* visited = new bool[V];
    for (int i = 0; i < V; ++i)
        visited[i] = false;

    stack<int> ST;
    vector<int>::iterator it;

    for(int i = 0; i < V; i++)
        if(!visited[i]) {
            cout << "xis wvwro-" << i << endl;
            ST.push(i);
            visited[i] = true;
            while(!ST.empty()) {
                int m= ST.top();
                ST.pop();

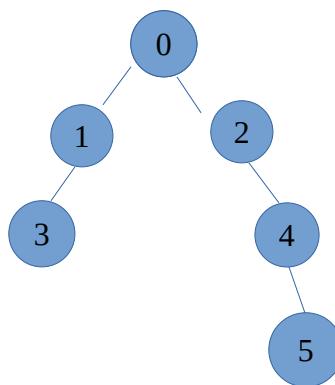
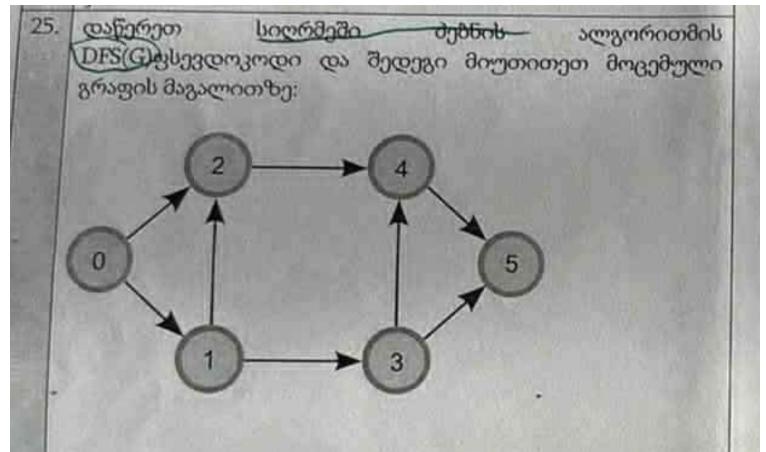
                for(it = adj[m].begin(); it < adj[m].end(); ++it)
                    if(!visited[*it]) {
                        ST.push(*it);
                        visited[*it] = true;
                        cout << m << "->" << *it << endl;
                    }
            }
        }
    }
}

```

visit

0	1	2	3	4	5
1	1	1	1	1	1

Stack
3
5
4
2
1
0



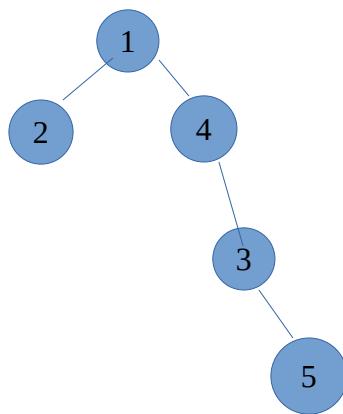
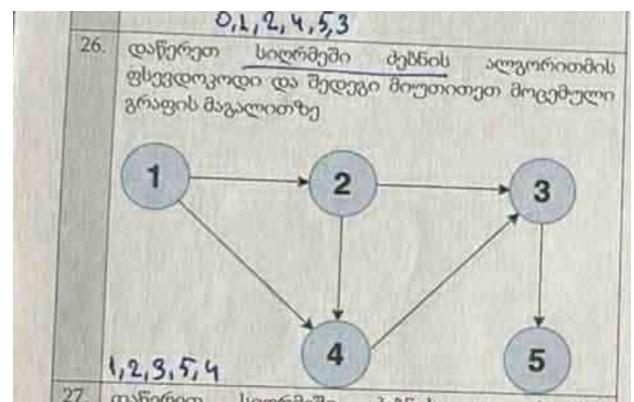
26.

visit

1	2	3	4	5
1	1	1	1	1

stack

5
3
4
2
1

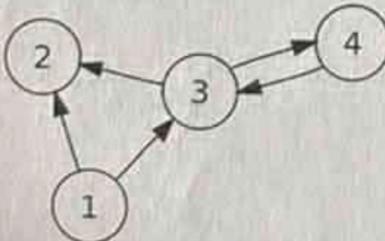


27.

visit

1	2	3	4
1	1	1	1

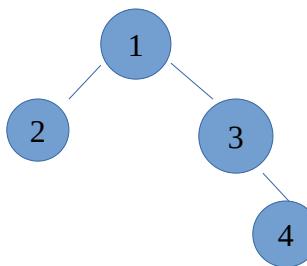
27. დაწერეთ სილომეში მებნის ალგორითმის
ფსევდოკოდი და შედეგი მიუთითეთ მოცემული
გრაფის მაგალითზე



1,2,3,4

Stack

4
3
2
1



28.

visit

A	B	C	D	E	F
1	1	1	1	1	1

Stack

D
F
E
C
B
A

