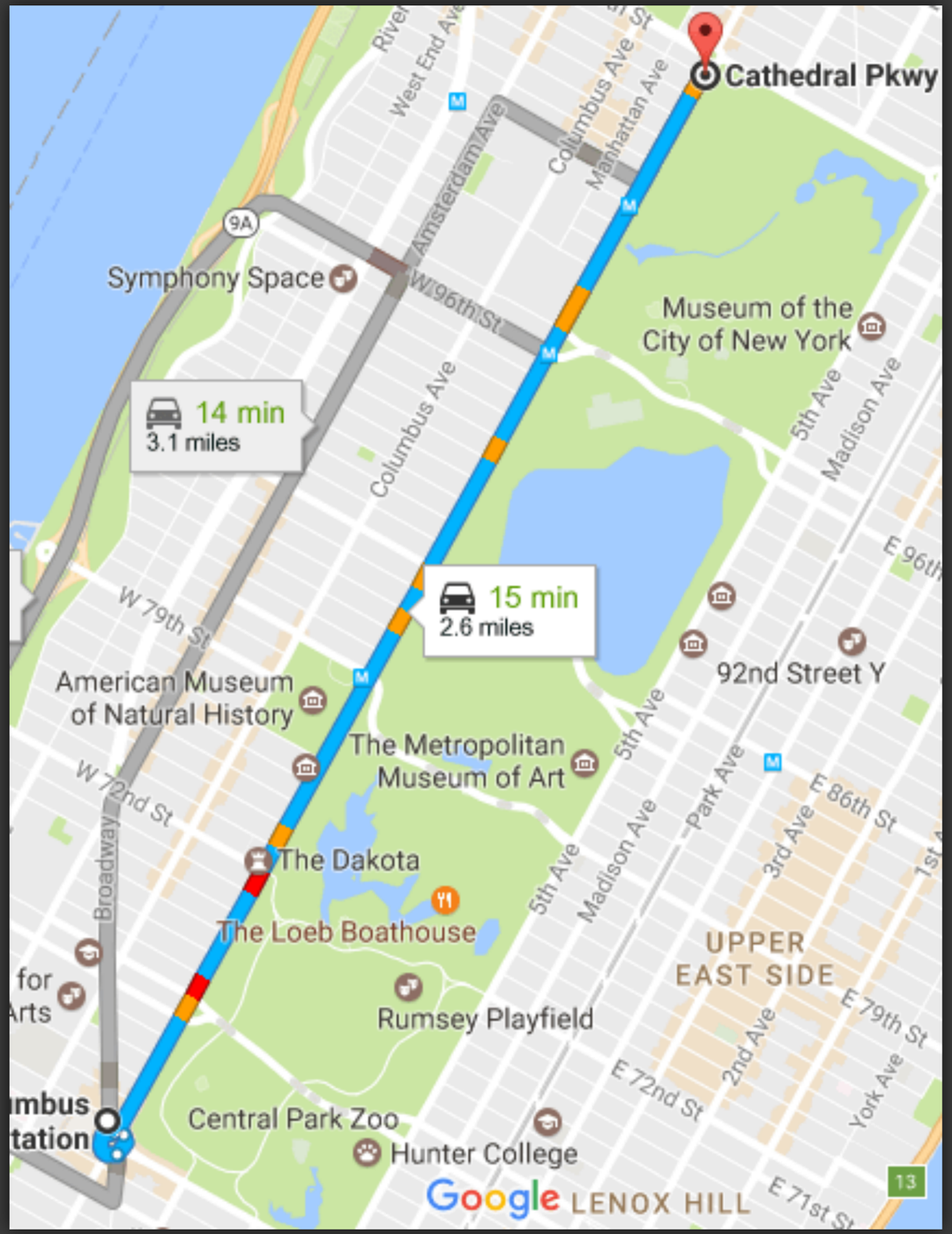
Introduction to Artificial Intelligence

AI 102

Saeed Mostafa Zalat | Teaching Assistant

College of Computer Science and Artificial intelligence

Pharos university in Alexandria



Blind Search Algorithms

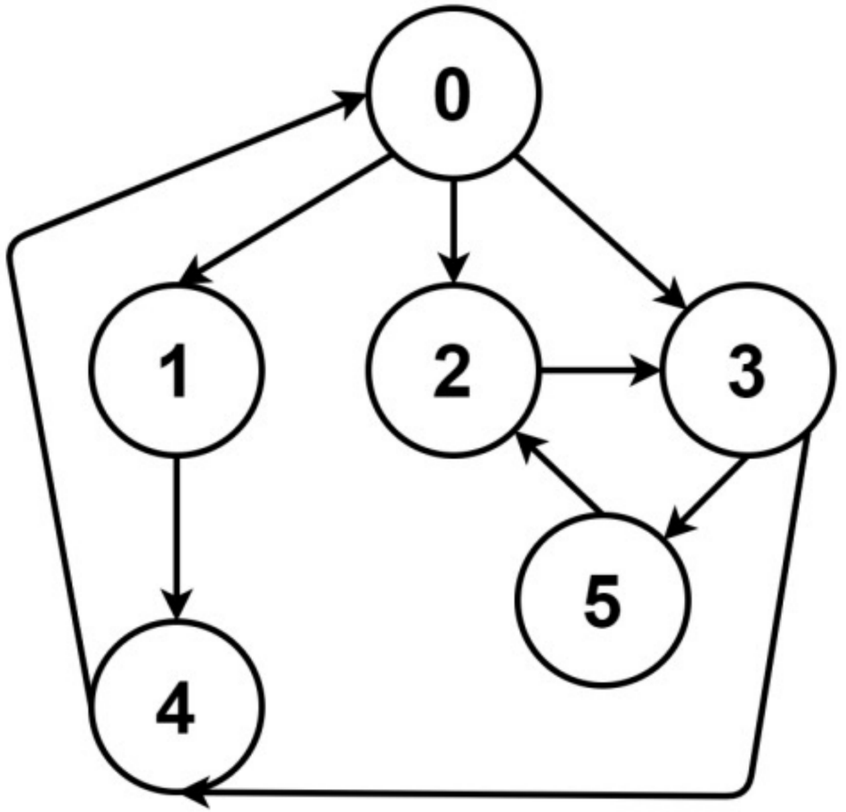
uninformed search

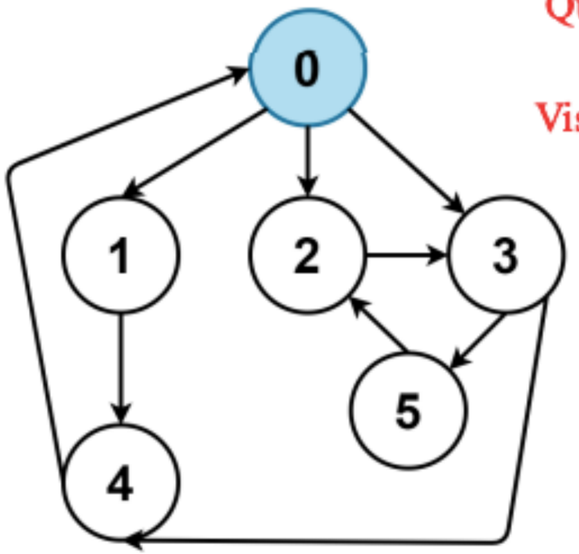
Graph traversal (Graph search) refers to the process of visiting each vertex in a graph.

Breadth First Search (BFS) is an algorithm traverses a graph using a **queue**.

Depth First Search (DFS) is an algorithm traverses a graph using a **stack**.

Directed Graph



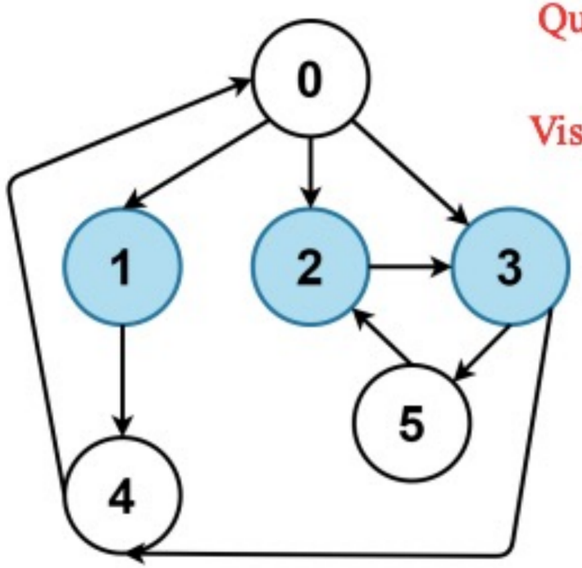


Queue

0									

Visited

0									



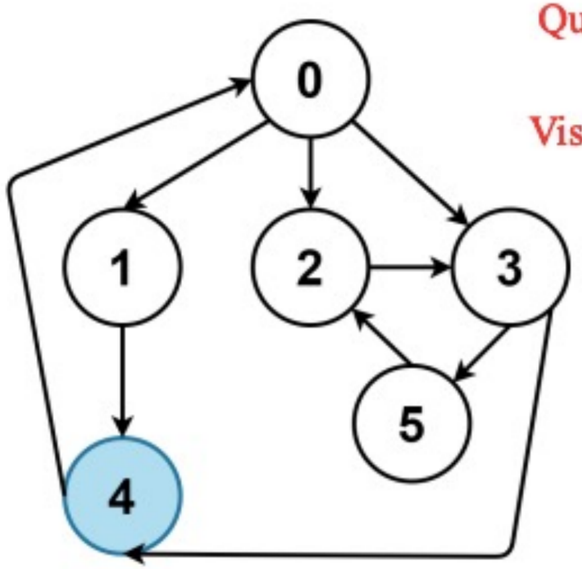
Queue

1	2	3							

Visited

0	1	2	3						

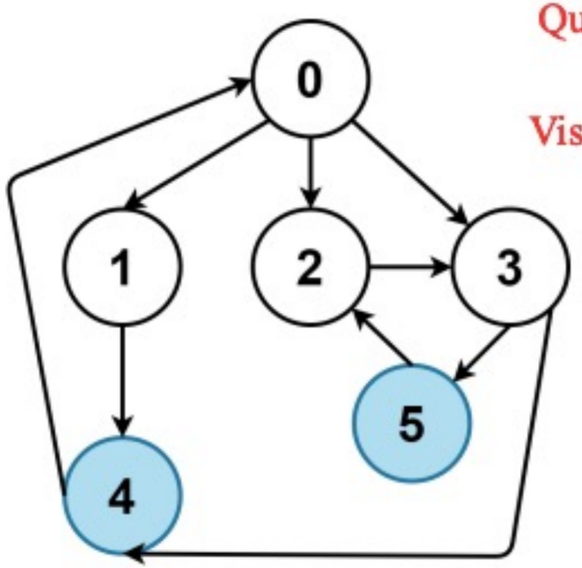
```
Printed: 0
```



Queue	2	3	4							

Visited	0	1	2	3	4					

```
Printed: 0, 1
```



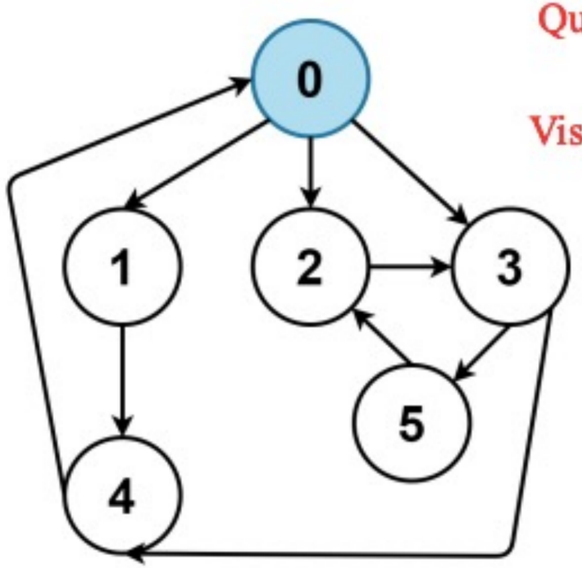
Queue

4	5								

Visited

0	1	2	3	4	5				

```
Printed: 0, 1, 2, 3
```



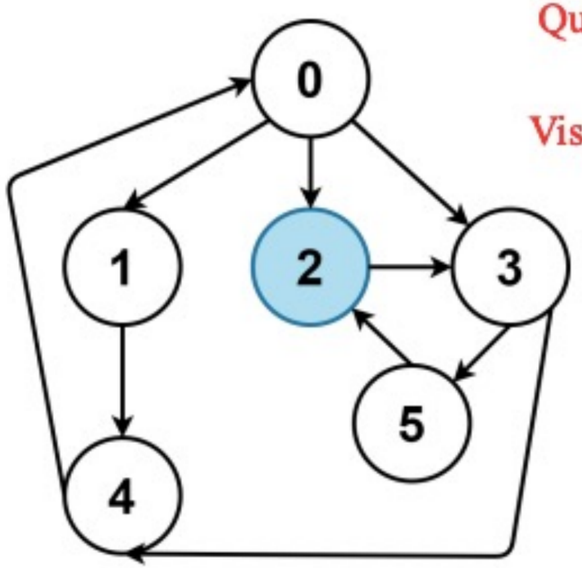
Queue

5									

Visited

0	1	2	3	4	5				

```
Printed: 0, 1, 2, 3, 4
```



Queue

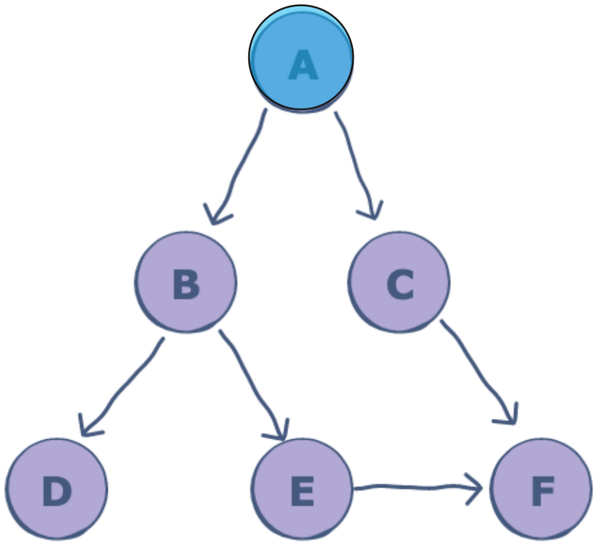
Visited

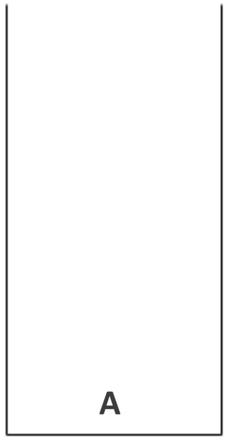
0	1	2	3	4	5				

```
Printed: 0, 1, 2, 3, 4, 5
```

```
'C' : ['F', 'G'],
'D' : [],
'E' : [],
'F' : [],
'G' : []
}
visited = []
queue = []
goal = 'F'

def bfs (visited, graph,node):
    visited.append(node)
    queue.append(node)
    
    while queue:
        s = queue.pop(0)
        print(s, end=" ")
        for neighbour in graph[s]:
            if neighbour not in visited:
                visited.append(neighbour)
                queue.append(neighbour)
            if goal in visited:
                break
bfs(visited, graph, 'A')
```

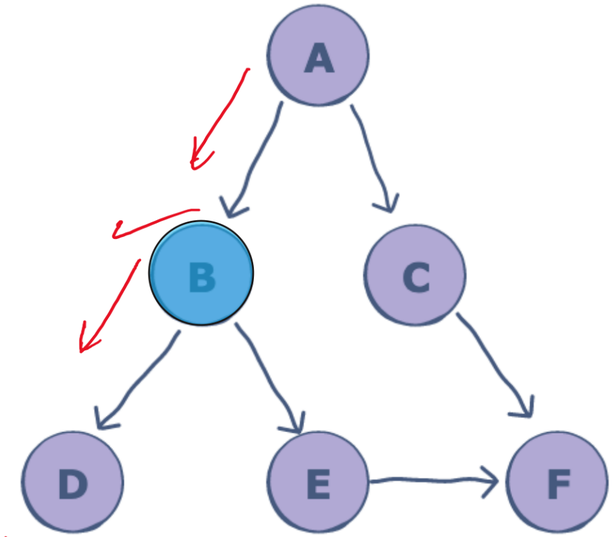


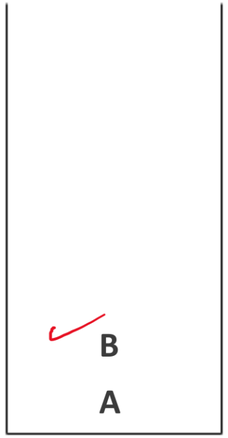


Visited

A									

Printed: A ,

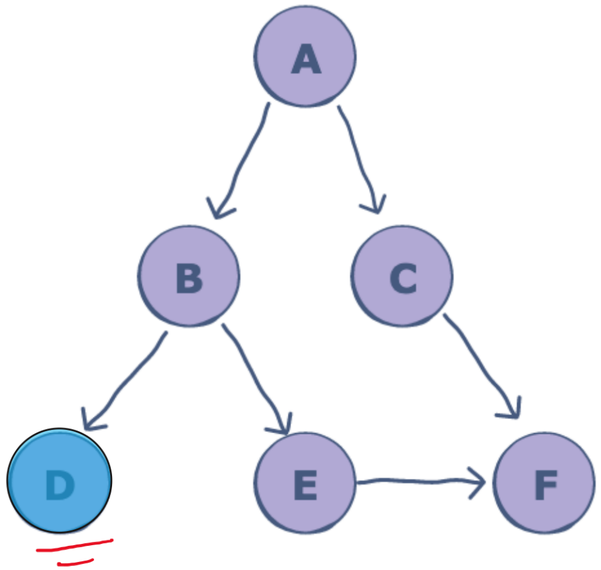


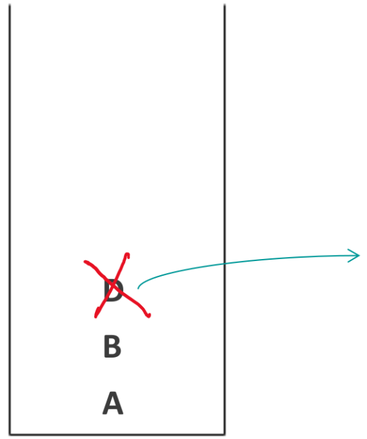


Visited

A	B								

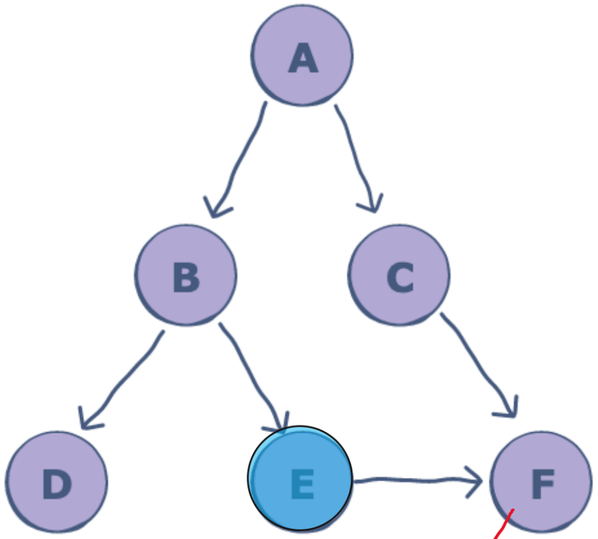
Printed: A , B

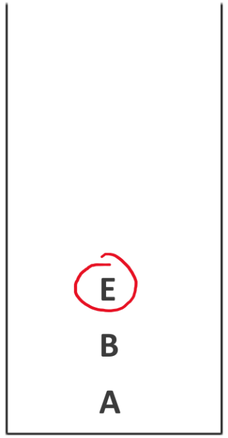




Visited	A	B	D							

Printed: A , B , D

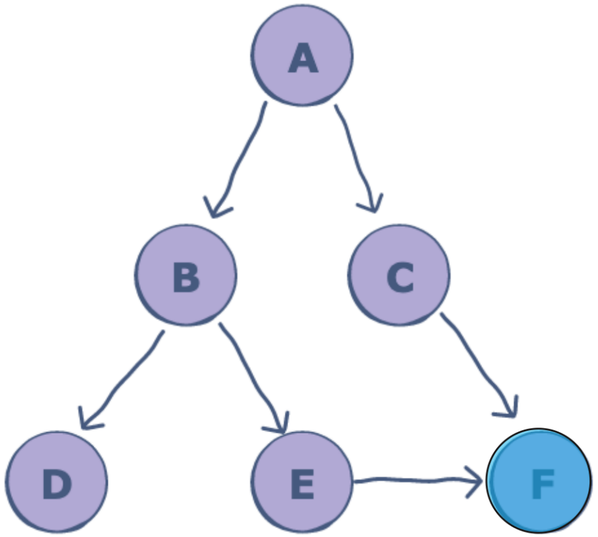


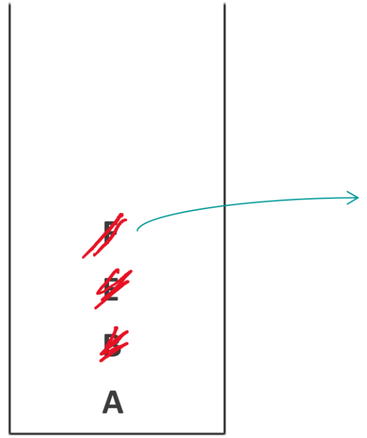


Visited

A	B	D	E						

Printed: A , B , D , E ,

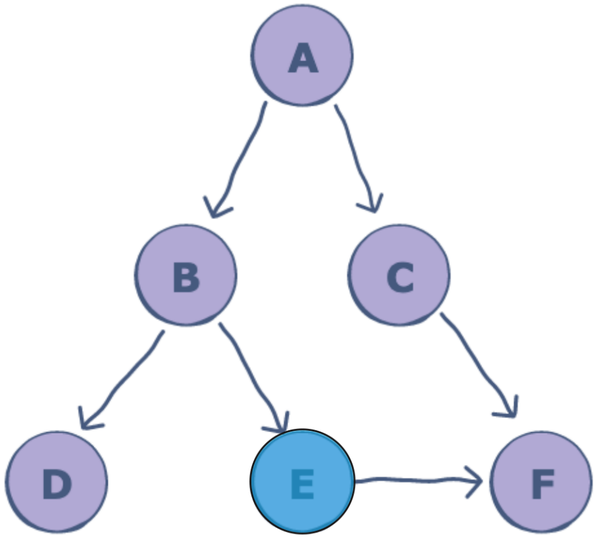


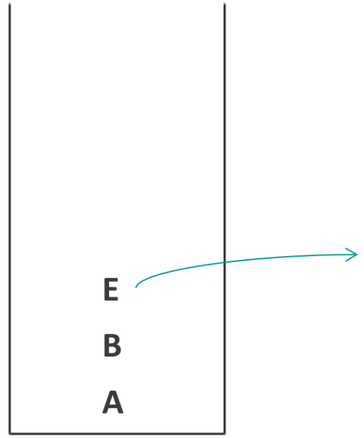


Visited

A	B	D	E	F					

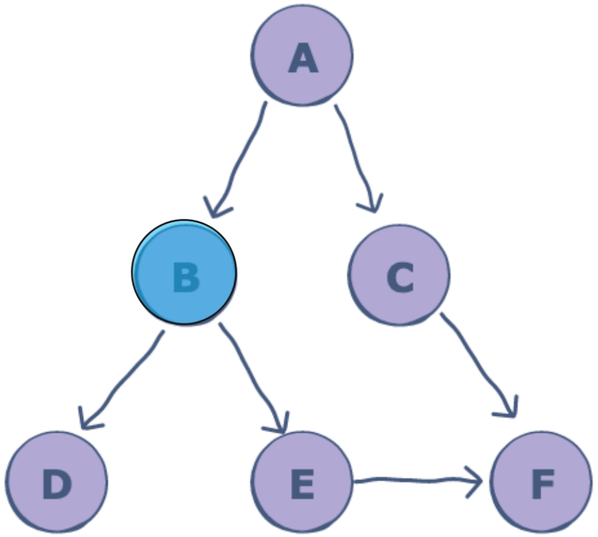
Printed: A , B , D , E , F ,

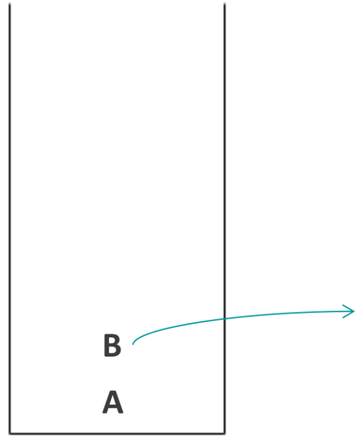




Visited	A	B	D	E	F					

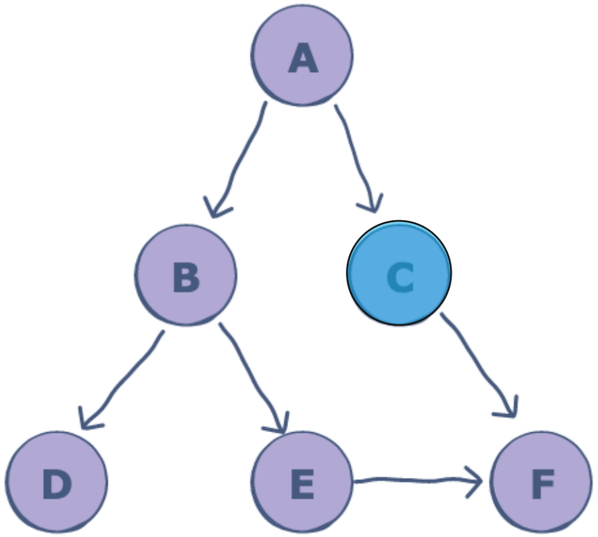
Printed: A , B , D , E , F ,

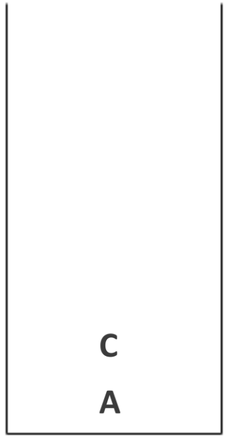




Visited	A	B	D	E	F					

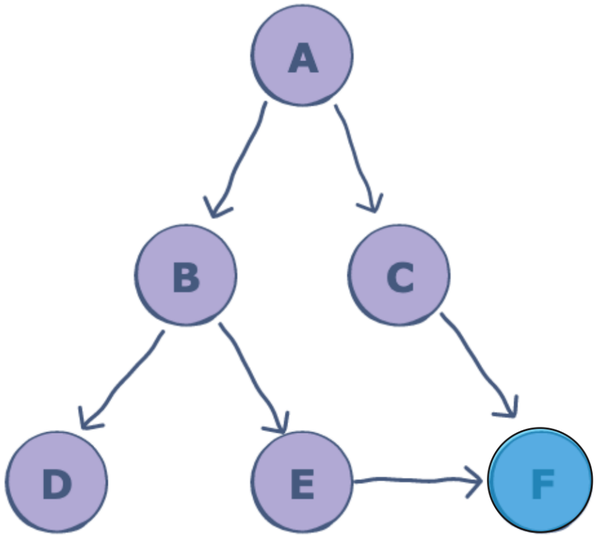
Printed: A , B , D , E , F ,

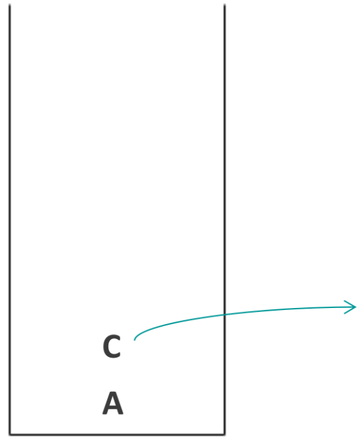




Visited	A	B	D	E	F	C				

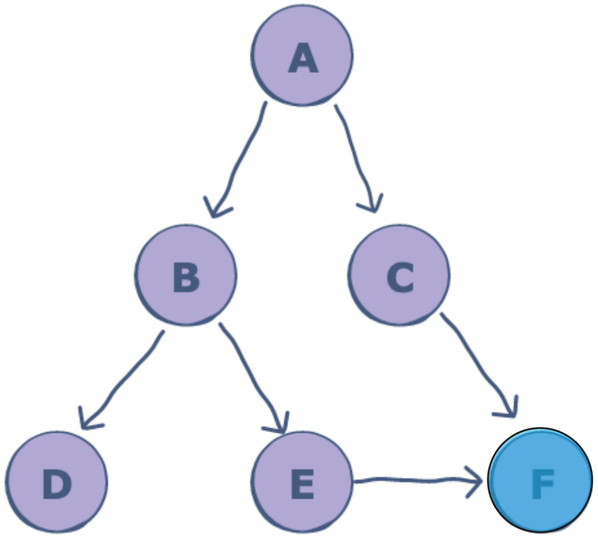
Printed: A , B , D , E , F , C

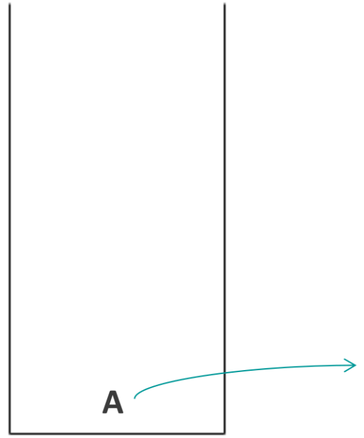




Visited	A	B	D	E	F	C				

Printed: A , B , D , E , F , C



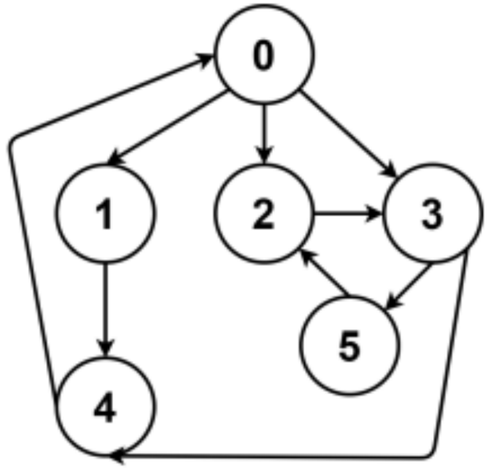


Visited	A	B	D	E	F	C				

Printed: A , B , D , E , F , C

A graph is represented in python using dictionary and list.

Graph Visualization



Graph Representation in Python

```
{
    0: [1, 2, 3],
    1: [4],
    2: [3],
    3: [4, 5],
    4: [0],
    5: [2]
}
```

```
graph = {
'A' : ['B','C'],
'B' : ['D', 'E'],
'C' : ['F', 'G'],
'D' : [],
'E' : [],
'F' : [],
'G' : []
}
goal = 'F'
visited = set()
def dfs (visited, graph, node):
    if node not in visited:
        print(node)
        visited.add(node)
    for neighbour in graph[node]:
        if goal in visited:
            break
        else:
            dfs(visited, graph, neighbour)

dfs(visited, graph, 'A')
```