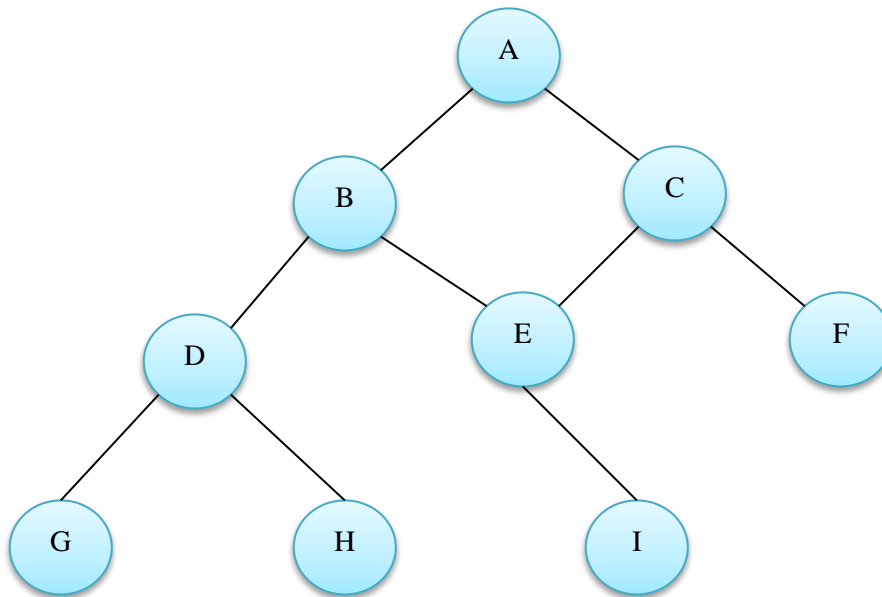


## Practical No. 1

Aim: (A) Implement Breath First Search algorithm.

(B) Implement Depth First Search algorithm.



(A) Implement Breath First Search algorithm.

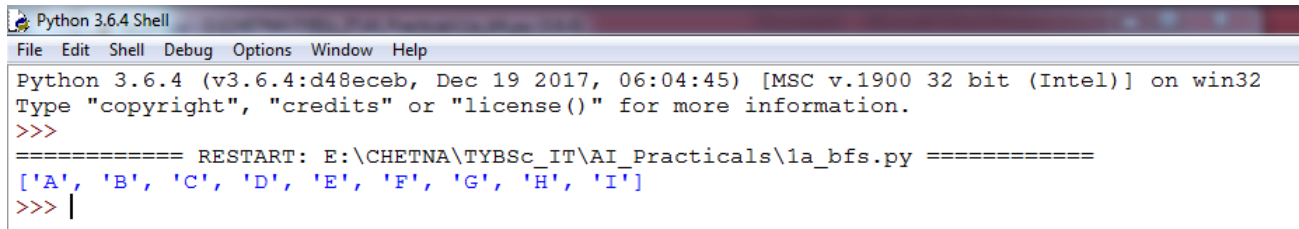
```
def bfs(graph, node, path=[], index=0):  
    path+= [node]  
    try:  
        while(True):  
            for neighbour in graph[path[index]]:  
                if neighbour not in path:  
                    path+= [neighbour]  
                    index+=1  
    except IndexError:  
        return path  
graph1={  
    'A': ['B', 'C'],  
    'B': ['A', 'D', 'E'],  
    'C': ['A', 'E', 'F'],  
    'D': ['B', 'G', 'H'],  
    'E': ['B', 'C', 'I'],  
    'F': ['C'],
```

```

        'G': ['D'],
        'H': ['D'],
        'I': ['E']
    }
    print(bfs(graph1,'A'))

```

Output:



```

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\CHETNA\TYBSc_IT\AI_Practicals\la_bfs.py =====
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']
>>> |

```

(B) Implement Depth First Search algorithm.

```

def dfs(graph, node):
    stack = [node]
    path = []

    while stack:
        vertex = stack.pop()
        if vertex in path:
            continue
        path.append(vertex)
        for neighbor in graph[vertex]:
            if neighbor not in path:
                stack.append(neighbor)

    return path

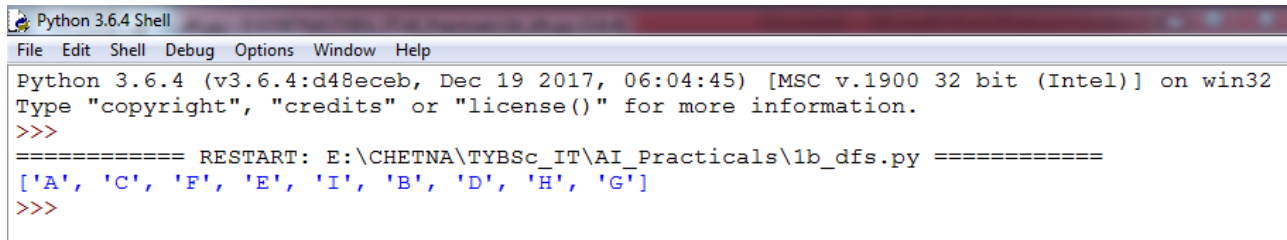
graph1 = {'A': ['B', 'C'],
          'B': ['A', 'D', 'E'],
          'C': ['A', 'E', 'F'],
          'D': ['B', 'G', 'H'],
          'E': ['B', 'C', 'I'],
          'F': ['C'],
          'G': ['D'],

```

```
'H': ['D'],  
'I': ['E']  
}
```

```
print(dfs(graph1, 'A'))
```

Output:



```
Python 3.6.4 Shell  
File Edit Shell Debug Options Window Help  
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: E:\CHETNA\TYBSc_IT\AI_Practicals\1b_dfs.py =====  
['A', 'C', 'F', 'E', 'I', 'B', 'D', 'H', 'G']  
>>>
```