

PRACTICAL 1: Breadth First Search

1. With set():

```
from collections import deque

graph1 = {
    'A':set(['B', 'C']),
    'B':set(['A', 'D', 'E']),
    'C':set(['A', 'F']),
    'D':set(['B']),
    'E':set(['B', 'F']),
    'F':set(['C', 'E'])
}

def bfs(graph ,start):
    visited,queue = [],deque([start])
    while queue:
        node = queue.popleft()
        if node not in visited:
            visited.append(node)
            queue.extend(graph[node] - set(visited))
    return visited
print(bfs(graph1,'A'))
```



```
>>>
['A', 'C', 'B', 'F', 'D', 'E']
>>>
```

PRACTICAL 1: Breadth First Search

2. Without set():

```
from collections import deque

graph1 = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}

def bfs(graph ,start):
    visited,queue = [],deque([start])
    while queue:
        node = queue.popleft()
        if node not in visited:
            visited.append(node)
            for neighbor in sorted (graph[node]):
                if neighbor not in visited and neighbor not in queue:
                    queue.append(neighbor)
    return visited
print(bfs(graph1,'A'))
```



```
>>>
['A', 'B', 'C', 'D', 'E', 'F']
>>>
```