

Lab 6**1. Write code for DLS**

```
graph = {
    'A':['B', 'C'],
    'B':['D', 'E'],
    'C': ['F'],
    'D': ['G','H'],
    'E': [],
    'F':['I','K'],
    'G':[],
    'H':['L'],
    'I':[],
    'K':['M'],
    'L':[],
    'M':[]
}

def dls(start,goal,path,lavel,max):
    print(lavel)
    print(start)
    path.append(start)
    if start==goal:
        return path
    if lavel==max:
        return path
    for child in graph[start]:
        if dls(child,goal,path,lavel+1,max):
            return path
    path.pop()
    return False

start='A'
goal=int(input("enter the goal state"))
max=int(input("enter the limit"))
path=list()
res=dls(start,goal,path,0,max)
if res:
    print(path)
```

Lab 6 Task

- 1. BFS without Queue & without Node**
- 2. BFS with Queue & Node**