

```

In [2]: SuccList={'a':['b','c'],'b':['a','c','d'],'c':['a','b','d'],'d':['b','c']}
Start='a'
Goal='d'
Closed = list()
SUCCESS=True
FAILURE=False
State=FAILURE

def GOALTEST(N):
    if N == Goal:
        return True
    else:
        return False

def MOVEGEN(N):
    New_list=list()
    if N in SuccList.keys():
        New_list=SuccList[N]
    print("New_list=",New_list)
    return New_list

def APPEND(L1,L2):
    New_list=L1+L2
    return New_list

def BFS():
    OPEN=[Start]
    CLOSED=list()
    global State
    global Closed
    while (len(OPEN) != 0) and (State != SUCCESS):
        print("-----")
        N= OPEN[0]
        print("N=",N)
        del OPEN[0] #delete the node we picked
        if GOALTEST(N)==True:
            State = SUCCESS
            CLOSED = APPEND(CLOSED,list(N))
            print("CLOSED=",CLOSED)
        else:
            CLOSED = APPEND(CLOSED,list(N))
            print("CLOSED=",CLOSED)
            CHILD = MOVEGEN(N)
            print("CHILD=",CHILD)
            for val in CLOSED:
                if val in CHILD:
                    CHILD.remove(val)

            for val in OPEN:
                if val in CHILD:
                    CHILD.remove(val)

            OPEN = APPEND(OPEN,CHILD) #append movegen elements to OPEN
            print("OPEN=",OPEN)

    Closed=CLOSED
    return State

```

```
#Driver Code
```

```
result=BFS() #call search algorithm  
print(Closed,result)
```

```
-----
```

```
N= a  
CLOSED= ['a']  
New_list= ['b', 'c']  
CHILD= ['b', 'c']  
OPEN= ['b', 'c']
```

```
-----
```

```
N= b  
CLOSED= ['a', 'b']  
New_list= ['a', 'c', 'd']  
CHILD= ['a', 'c', 'd']  
OPEN= ['c', 'd']
```

```
-----
```

```
N= c  
CLOSED= ['a', 'b', 'c']  
New_list= ['a', 'b', 'd']  
CHILD= ['a', 'b', 'd']  
OPEN= ['d']
```

```
-----
```

```
N= d  
CLOSED= ['a', 'b', 'c', 'd']  
['a', 'b', 'c', 'd'] True
```

```
In [ ]:
```