**Best First Search**

SuccList ={ 'S':[['A',3],['B',6],['C',5]], 'A':[['E',8],['D',9]],'B':[['G',14],['F',12]], 'C':[['H',7]], 'H':[['J',6],['I',5]],'I': [['M',2],['L',10],['K',1]]} *#Graph(Tree) List*  
  
Start= input("Enter Source node >> ").upper()  
Goal= input('Enter Goal node >> ').upper()  
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]  
   
    **return** New\_list  
   
**def** APPEND(L1,L2):  
    New\_list=list(L1)+list(L2)  
    **return** New\_list  
   
**def** SORT(L):  
    L.sort(key = **lambda** x: x[1])   
    **return** L   
**def** BestFirstSearch():  
    OPEN=[[Start,5]]  
    CLOSED=list()  
    **global** State  
    **global** Closed  
    i=1  
    **while** (len(OPEN) != 0) **and** (State != SUCCESS):  
        print("\n<<<<<<<<<<---({})--->>>>>>>>>>\n".format(i))  
        N= OPEN[0]  
        print("N=",N)  
        **del** OPEN[0] *#delete​ the node we picked*  
        **if** GOALTEST(N[0])==True:  
            State = SUCCESS  
            CLOSED = APPEND(CLOSED,[N])  
            print("CLOSED=",CLOSED)  
        **else**:  
            CLOSED = APPEND(CLOSED,[N])  
            print("CLOSED=",CLOSED)  
            CHILD = MOVEGEN(N[0])  
            print("CHILD=",CHILD)  
            **for** val **in** OPEN:  
                **if** val **in** CHILD:               
                    CHILD.remove(val)  
            **for** val **in** CLOSED:  
                **if** val **in** CHILD:             
                    CHILD.remove(val)  
            OPEN = APPEND(CHILD,OPEN) *#append​ movegen elements to OPEN*  
            print("Unsorted OPEN=",OPEN)  
            SORT(OPEN)  
            print("Sorted OPEN=",OPEN)  
            Closed=CLOSED  
            i+=1  
    **return** State  
*#code by <<<Sahil Gaonkar>>>*  
result=BestFirstSearch()  
print("Best First Search Path >>>> {} <<<{}>>>".format(Closed, result))

**OUTPUT:**

Enter Source node >> S

Enter Goal node >> G

<<<<<<<<<<---(1)--->>>>>>>>>>

N= ['S', 5]

CLOSED= [['S', 5]]

CHILD= [['A', 3], ['B', 6], ['C', 5]]

Unsorted OPEN= [['A', 3], ['B', 6], ['C', 5]]

Sorted OPEN= [['A', 3], ['C', 5], ['B', 6]]