

UCS

```
import copy

def uniform_cost_search(graph,start,goal):

    path=[]

    visited=[start]

    path_cost=0    #initially the cost will be 0

    if start==goal:

        return path,path_cost,visited

    path.append(start)

    openlist=[(path_cost,path)]

    while len(openlist)>0:

        currcost,currpath=openlist.pop(0)

        print('The current path is',currpath)

        currnode=currpath[-1]

        if currnode==goal:

            return currpath,currcost,visited

        if currnode not in visited:

            visited.append(currnode)

            neighbours=graph[currnode]

            print('The neighbours are',neighbours)

            for n in neighbours:

                n_path_cost=currcost+n[0]

                n_path=copy.copy(currpath)

                n_path.append(n[1])

                n_openlist_ele=(n_path_cost,n_path)

                if n[1] not in visited:

                    openlist.append(n_openlist_ele)

            openlist.sort()

            print('current openlist after appending',openlist)

        print("")

    return path, n_path_cost, visited
```

```
graph2={0:[(1,2),(1,1)],2:[(2,5)],1:[(3,3)],3:[(2,5),(2,4)],4:[(1,5)],5:[(3,0)]}  
ufs=uniform_cost_search(graph2,0,4)  
print(ufs)
```