

```
# ****均一コスト探索 ****
```

```
# g()関数はstartからa_nodeまでのcostコストを計算する
def g(graph, path, a_node):
    cost = 0
    print(".... path[0]:",path[0])
    start,c = path[0]
    path = path[1:]
    path = [n for n,v in path]
    path = path +[a_node] #[(a_node,graph[start][a_node])]
    print(".... start:",start)
    print(".... path:",path)
    for node in path:
        cost += graph[start][node]
        start = node
    return cost
```

```
#均一コスト探索アルゴリズムのプログラム
```

```
def UCS(graph,start,goal):
    O=[]
    C=[]
    cost = 0
    initPath = [(start,cost)] # Path init
    O.append(start)
    pathPriorityQueue = [initPath] # List of paths
    while len(pathPriorityQueue) != 0:
        tmpPath = pathPriorityQueue.pop(0)
        print("tmpPath=",tmpPath)
        pnode = O.pop(0)
        if pnode not in C:
            C.append(pnode)
        if pnode == goal:
            print ("Goal! UCS search is over!")
            return tmpPath
        else:
            P_A = graph[pnode]
            print(".. P_A=",P_A)
            pq=[]
            for node in P_A:
                print("... node =",node)
                if node not in O and node not in C:
                    cost = g(graph,tmpPath,node)
                    O.append(node)
                    print("... OpenList:",O)
                    print("... ClosedList:",C)
                    pq += [(node,cost)]
            if pq == []:
                break
            pq = sorted(pq,key=lambda x:x[1])
            print(".. pq =",pq)
            O=[n for n,v in pq]
            if goal in O[1:]:
                O.remove(goal)
            newPath = tmpPath + [pq[0]]
            pathPriorityQueue.append(newPath)
```

```
# メインプログラム：グラフの定義
# A->D:6, A->C:2, ....
```

```
g2 = {'A':{'D':6,'C':2}, 'B':{'A':3,'D':8}, 'C':{'D':7,'E':5}, 'D':{'E':-2}}
```

```
graph3 = {'A':{'B':3,'E':9,'C':2},
          'B':{'D':2,'E':4},
          'C':{'E':6,'F':9},
          'E':{'G':1,'H':2},
          'D':{'G':3},
          'F':{'H':1,'I':2},
          'G':{'J':5},
          'H':{'J':5,'L':9,'K':6},
          'I':{'K':2},
          'K':{'L':3},
          'J':{'L':5}}
```

```
# メインプログラム：与えた問題（グラフ、スタート、ゴール）の解の探索
```

```
print("***** Uniform Cost Search for graph g2 :*****")
UCS(g2, 'B', 'E')
```

```
print("***** Uniform Cost Search for graph graph3 :*****")
UCS(graph3, 'A', 'L')
```