Program 1

In [13]:

```
from heuristicsearch.a_star_search import AStar
print('Graph-A')
adj_list  ={
    'A': [('B',2),('E',3)],
    'B': [('C',1),('G',9)],
    'C':None,
    'E': [('D',6)],
    'D': [('G',1)],
}

Heuristic = {
        'A':10,
        'B':6,
        'C':99,
        'D':1,
        'E':7,
        'G':0,
    }

graph=AStar(adj_list,Heuristic)
graph.apply_a_star(start='A',stop='G')
```

```
Graph-A
Path
A -> E -> D -> G
Cost
0 -> 3 -> 9 -> 10
```

Program 2

In [13]:

```
from heuristicsearch.ao_star import AOStar
print('Graph-A')

adj_list  ={
 'A': [[('C',2), ('D',3)], [('B',4)]],
 'B': [[('E',1)], [('F',4)]],
 'C': [[('G',3)], [('H',2), ('I',3)]],
 'D': [ [('J',3)]]
}


Heuristic ={
 'A': -1,
 'B': 4,
 'C': 2,
 'D': 3,
 'E': 6,
 'F': 8,
 'G': 2,
 'H': 0,
 'I': 0,
 'J': 0
}

graph=AOStar(adj_list,Heuristic,'A')
graph.applyAOStar()
```

```
Graph-A
PROCESSING NODE : A
-----------------------------------------------------------------------------------------
8 ['B']

PROCESSING NODE : B
-----------------------------------------------------------------------------------------
7 ['E']

PROCESSING NODE : A
-----------------------------------------------------------------------------------------
10 ['C', 'D']

PROCESSING NODE : E
-----------------------------------------------------------------------------------------
0 []

PROCESSING NODE : B
-----------------------------------------------------------------------------------------
1 ['E']

PROCESSING NODE : A
-----------------------------------------------------------------------------------------
5 ['B']

FOR THE SOLUTION, TRAVERSE THE GRAPH FROM THE START NODE: A
------------------------------------------------------------
{'E': [], 'B': ['E'], 'A': ['B']}
------------------------------------------------------------
```

Program 4

In [16]:

```
from decisiontree.ID3Algorithm import ID3
import csv
```

In [17]:

```
def load_csv(filename):
    lines=csv.reader(open(filename,"r"))
    dataset = list(lines)
    headers = dataset.pop(0)
    return dataset,headers
```

In [18]:

```
dataset_train, headers_train = load_csv("F:\data.csv")
dataset_test, headers_test = load_csv("F:\data.csv")
```

In [19]:

```
id3 = ID3(dataset_train,headers_train,dataset_test,headers_test)
```

In [20]:

```
id3.build_tree()
```

```
The decision tree for the dataset using ID3 algorithm is
outlook
├── overcast
│   └── yes
├── rainy
│   └── wind
│       ├── strong
│       │   └── no
│       └── weak
│           └── yes
└── sunny
    └── humidity
        ├── high
        │   └── no
        └── normal
            └── yes
```

In [21]:

```
id3.classify()
```

```
The test instance: ['sunny', 'hot', 'high', 'weak', 'no']
The label for test instance:
no
The tree traversal for the test instance is:
outlook
└── sunny
    └── humidity
        └── high
            └── no

The test instance: ['sunny', 'hot', 'high', 'strong', 'no']
The label for test instance:
no
The tree traversal for the test instance is:
outlook
└── sunny
    └── humidity
        └── high
            └── no
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: