A Star algorithm using built in Library

(Install heuristic search- Command: **pip install heuristicsearch)**

```
from heuristicsearch.a_star_search import AStar

graph_nodes = {
    'A': [('B', 6), ('F', 3)],
    'B': [('C', 3), ('D', 2)],
    'C': [('D', 1), ('E', 5)],
    'D': [('C', 1), ('E', 8)],
    'E': [('I', 5), ('J', 5)],
    'F': [('G', 1),('H', 7)] ,
    'G': [('I', 3)],
    'H': [('I', 2)],
    'I': [('E', 5), ('J', 3)],

}

heuristics = {
            'A': 10,
            'B': 8,
            'C': 5,
            'D': 7,
            'E': 3,
            'F': 6,
            'G': 5,
            'H': 3,
            'I': 1,
            'J': 0
        }
graph= AStar(graph_nodes,heuristics)
graph.apply_a_star(start='A', stop='J')
```

OUTPUT:

```
Path
A -> F -> G -> I -> J
Cost
0 -> 3 -> 4 -> 7 -> 10
```

AO* Star algorithm using built in Library

```
from heuristicsearch.ao_star import AOStar
print("Graph-1")

heuristic = {'S': 1, 'A': 7, 'B': 12, 'C': 13, 'D': 5, 'E': 6, 'F': 5, 'G': 7, 'H': 2,}

adjacency_matrix = {
    'S': [[('A', 1), ('B', 1)], [('C', 1)]],
    'A': [[('D', 1)], [('E', 1)]],
    'C': [[('F', 1), ('G', 1)]],
    'D': [('H', 1)]
}
graph=AOStar(adjacency_matrix,heuristic,'S')
graph.applyAOStar()
```

Output:

```
Graph-1
PROCESSING NODE : S
--------------------------------------------------------------------------
14 ['C']

PROCESSING NODE : C
--------------------------------------------------------------------------
14 ['F', 'G']

PROCESSING NODE : S
--------------------------------------------------------------------------
15 ['C']

PROCESSING NODE : F
--------------------------------------------------------------------------
0 []

PROCESSING NODE : C
--------------------------------------------------------------------------
9 ['F', 'G']

PROCESSING NODE : S
--------------------------------------------------------------------------
10 ['C']

PROCESSING NODE : G
--------------------------------------------------------------------------
0 []

PROCESSING NODE : C
```

```
--------------------------------------------------------------------------
2 ['F', 'G']


PROCESSING NODE : S
--------------------------------------------------------------------------
3 ['C']

FOR THE SOLUTION, TRAVERSE THE GRAPH FROM THE START NODE: S
------------------------------------------------------------
{'F': [], 'G': [], 'C': ['F', 'G'], 'S': ['C']}
------------------------------------------------------------
```