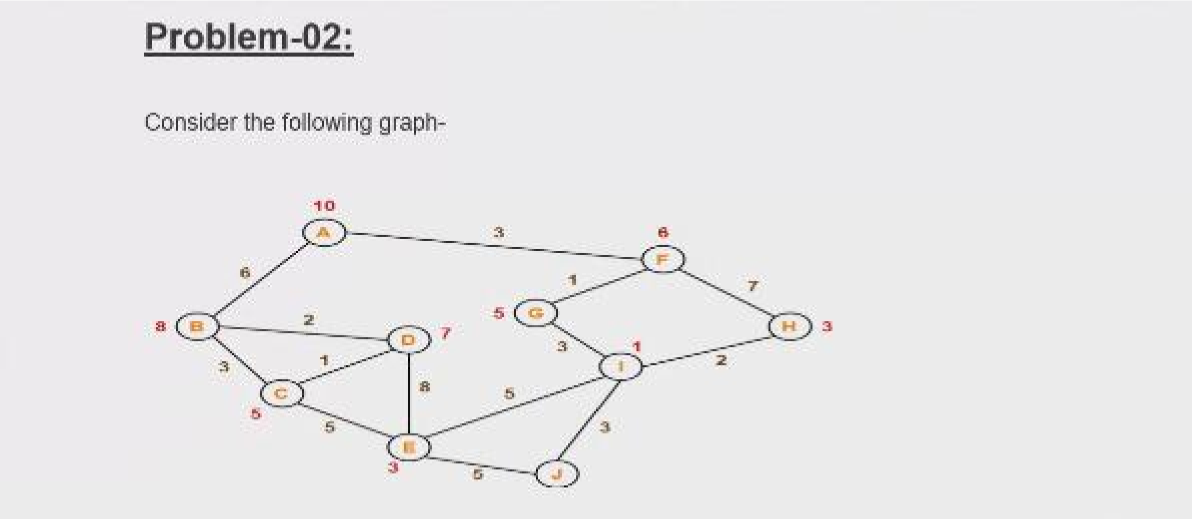
A Star



Problem-02:

Consider the following graph-

In [1]:
```
graph = {
    'A': [('F', 3), ('B', 6)],
    'B': [('C', 3),('D',2)],
    'C': [('D', 1),('E',5),('B',3)],
    'D': [('B',2),('E',8),('C',1)],
    'E': [('I',5),('J',5),('D',8),('C',5)],
    'F': [('G',1),('H',7)],
    'G': [('I',3),('F',1)],
    'H': [('I',2),('F',7)],
    'I': [('G',3),('H',2),('E',5),('J',3)],
    'J': [('E',5),('I',3)]
}
h_val = {'A':10,'B':8,'C':5,'D':7,'E':3,'F':6,'G':5,'H':3,'I':1,'J':0}
```

In [2]:

```
def a_star(start, goal):
    open_list = set([start])
    closed_list = set([])

    distance = {}
    distance[start] = 0
    parents = {}
    parents[start] = start

    while len(open_list) > 0:
        n = None
        for v in open_list:
            if n == None or distance[v] + h_val[v] < distance[n] + h_val[n]:
                n = v;
        if n == None:
            print('No Path exists')
            return
        if n == goal:
            path = []
            while parents[n] != n:
                path.append(n)
                n = parents[n]
            path.append(start)
            path.reverse()
            print('Path found: {0} Cost:{1}'.format(path,distance[goal]))
            return
        for (m, weight) in graph[n]:
            if m not in open_list and m not in closed_list:
                open_list.add(m)
                parents[m] = n
                distance[m] = distance[n] + weight
            else:
                if distance[m] > distance[n] + weight:
                    distance[m] = distance[n] + weight
                    parents[m] = n
                    if m in closed_list:
                        closed_list.remove(m)
                        open_list.add(m)
        open_list.remove(n)
        closed_list.add(n)
    print('No Path exists')
    return
```

In [3]:

```
a_star('A', 'J')
```

```
Path found: ['A', 'F', 'G', 'I', 'J'] Cost:10
```

In [ ]: