```
"""
Lab-1
Implement A* Search Algorithm
"""

inf=99999
g=[
    [0,3,inf,inf,4,inf,inf,inf],
    [inf,0,4,inf,5,inf,inf,inf],
    [inf,inf,0,4,inf,5,inf,inf],
    [inf,inf,inf,0,inf,inf,inf,inf],
    [inf,inf,inf,inf,0,2,inf,inf],
    [inf,inf,inf,inf,inf,0,4,inf],
    [inf,inf,inf,inf,inf,inf,0,3.5],
    [inf,inf,inf,inf,inf,inf,inf,0],
   ]
h=[11.5,10.1,5.8,3.4,9.2,7.1,3.5,0]

src=0
goal=6

class obj:
    def __init__(self,cost,path):
        self.cost=cost
        self.path=path

arr=[]
new_item=obj(h[src],[src])
arr.append(new_item)

while arr:
    cur_item=arr[0]
    cur_node=cur_item.path[-1]
```

```
    cur_cost=cur_item.cost
    cur_path=cur_item.path
    for i in range(0,len(h)):
        if g[cur_node][i]!=inf and g[cur_node][i]!=0:
            new_cost=cur_cost-h[cur_node]+h[i]+g[cur_node][i]
            new_path=cur_path.copy()
            new_path.append(i)
            if i==goal:
                print(new_cost)
                print(new_path)
            new_item=obj(new_cost,new_path)
            arr.append(new_item)
    arr.pop(0)
    arr=sorted(arr,key=lambda item: item.cost)

"""
Output:

13.5
[0, 4, 5, 6]
17.5
[0, 1, 4, 5, 6]
19.5
[0, 1, 2, 5, 6]
"""
```