

Practical 5

Aim: Implement best first search.

Code:

```
from queue import PriorityQueue
```

```
v = 14
```

```
graph = [[] for i in range(v)]
```

```
def best_first_search(actual_Src, target, n):
```

```
    visited = [False] * n
```

```
    pq = PriorityQueue()
```

```
    pq.put((0, actual_Src))
```

```
    visited[actual_Src] = True
```

```
    while pq.empty() == False:
```

```
        u = pq.get()[1]
```

```
        print(u, end=" ")
```

```
        if u == target:
```

```
            break
```

```
        for v, c in graph[u]:
```

```
            if visited[v] == False:
```

```
                visited[v] = True
```

```
                pq.put((c, v))
```

```
    print()
```

```
def addedge(x, y, cost):  
    graph[x].append((y, cost))  
    graph[y].append((x, cost))
```

```
adddedge(0, 1, 3)  
adddedge(0, 2, 6)  
adddedge(0, 3, 5)  
adddedge(1, 4, 9)  
adddedge(1, 5, 8)  
adddedge(2, 6, 12)  
adddedge(2, 7, 14)  
adddedge(3, 8, 7)  
adddedge(8, 9, 5)  
adddedge(8, 10, 6)  
adddedge(9, 11, 1)  
adddedge(9, 12, 10)  
adddedge(9, 13, 2)
```

```
source = 0  
target = 7  
best_first_search(source, target, v)
```

output:

```
0 1 3 2 8 9 11 13 10 5 4 12 6 7  
> |
```