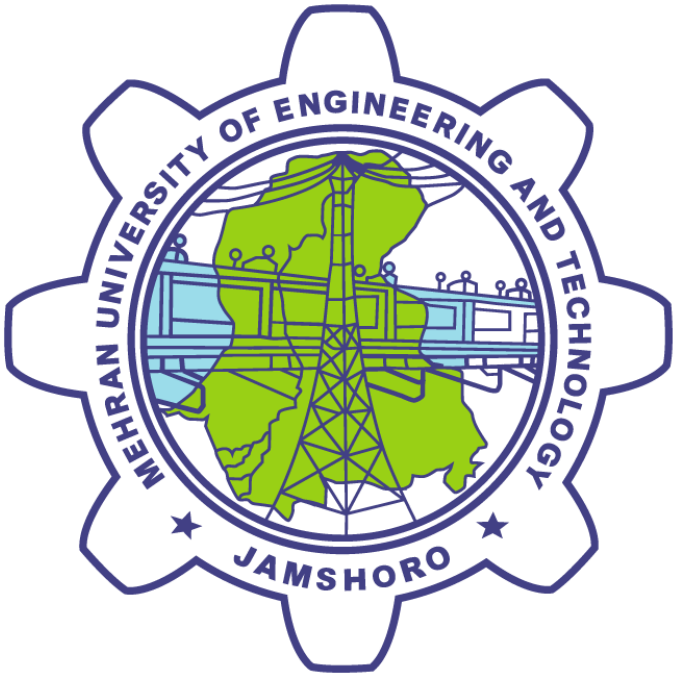

MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY
JAMSHORO

CPP PROJECT

Introduction:

The main idea of this project is to order product from the information provided. This project will focus on the development of a system for an online store that specializes in selling cameras. The system will be designed to streamline the process of managing inventory, processing orders, and handling customer information.

The project will utilize data structures and algorithms to efficiently manage the store's inventory and process orders. The system will also include features of a user-friendly interface for customers.

In addition to the features mentioned previously, this project will also incorporate the use of various data structures to enhance the efficiency and performance of the system.

Overall, this project will demonstrate the practical application of various data structures and algorithms in the design and development of an online shopping store management system that sells cameras, by providing efficient and optimized solutions for data storage, retrieval and pathfinding **Implemented Data Structures:**

- **Linked List:**
  Linked lists are a data structure that allows for efficient insertion and deletion of elements.
  Reason

The program uses linked lists to manage customer information and orders. The data of user (Delivery, Take Away) is saved in the linked List. We used Linked List instead of array because of dynamic size.

- **AVL (Adelson-Velsky and Landis) tree:**
  The AVL tree is a self-balancing binary search tree that is well-suited for storing large amounts of data and allows for fast insertion, deletion, and searching operations.
  Reason
  The AVL, for example, will be used to efficiently manage and maintain the store's inventory. All the operations of take away user is handled using AVL tree. The operations like searching, deletion, insertion is done in a better way. The insertion and deletion is done by order ID that is given by the user.

- **Graphs:**
  Graphs has been used in the project to show the areas where the parcel can be delivered. Graphs are represented through adjacency matrix.

```
Islamabad-----------------------
 { 0, 4, 0, 2, 0, 0 },
 { 4, 0, 3, 1, 0, 6 },   //I-8
 { 0, 3, 0, 4, 0, 7 },   //Askari
 { 2, 1, 4, 0, 9, 0 },   //F-10
 { 0, 0, 0, 9, 0, 5 },   //F-7
 { 0, 6, 7, 0, 5, 0 } }; //H-12
```

- **Dijkstra's algorithm:**

  Dijkstra's algorithm is used to implement efficient searching and navigation within the system. Dijkstra's algorithm is a shortest path algorithm that can be used to find the shortest path between two nodes in a graph. The nodes are represented as cities and this algorithm helps to provide the distance between cities.

  Reason

  This algorithm is often used in routing and as a subroutine in other graph algorithms. The nodes are represented as cities and this algorithm helps to provide the distance between cities.

```
void dijkstra(int graph[V][V], int dist[], int src)
{
    bool visited[V];

    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, visited[i] = false;

    dist[src] = 0;

    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, visited);

        visited[u] = true;

        for (int v = 0; v < V; v++)

            if (!visited[v] && graph[u][v] && dist[u] != INT_MAX && dist[u] + graph[u][v] < dist[v])
                dist[v] = dist[u] + graph[u][v];
    }
}
```

- **Prim’s algorithm:**
  Prim's algorithm can be used to find the minimum spanning tree of a graph. So, it provides the distances between areas in a city. **Reason**

  Prims is used between areas in a city. The reason for using this is that we wanted to give discount to user and previous distance is not used. The key idea behind Prim's algorithm is that at each step, we are adding a vertex to the tree that is connected to the tree by the minimumweight edge. This ensures that the tree will have the minimum total weight of any spanning tree that can be formed from the graph.

```
void prims(int graph[V1][V1], int distanceP[])
{
    int parent[V1];
    bool visitedP[V1];

    for (int i = 0; i < V1; i++)
        distanceP[i] = INT_MAX, visitedP[i] = false;

    distanceP[0] = 0;
    parent[0] = -1;

    for (int count = 0; count < V1 - 1; count++) {

        int u = minKey(distanceP, visitedP);

        visitedP[u] = true;

        for (int v = 0; v < V1; v++)

            if (graph[u][v] && visitedP[v] == false && graph[u][v] < distanceP[v])
                parent[v] = u, distanceP[v] = graph[u][v];
    }

}
```

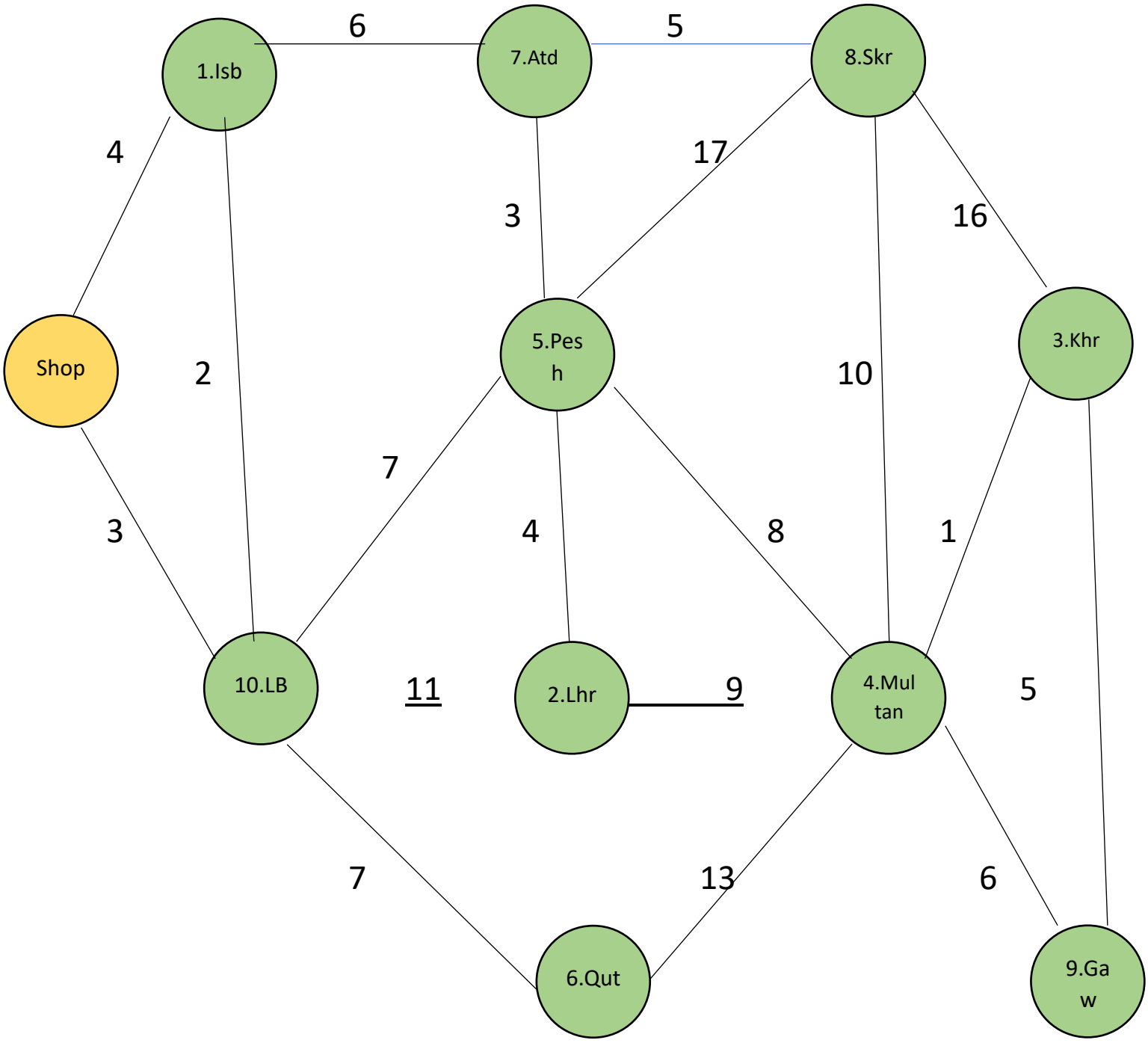
Product menu:

ITEM NO.	ITEM NAME	ORIGINAL PRICE
1	Sony FX30	45500
2	Sony FR7	175000
3	Canon EOS 90D	289000
4	Nikon D6	155700
5	Panasonic Lumix G10	75000
6	Canon EOS 250D	210000
7	Sony Alpha 7	199999
8	Nikon D750	100000
9	Panasonic Lumix GH5	45000
10	Canon EOS 5D Mark	500000

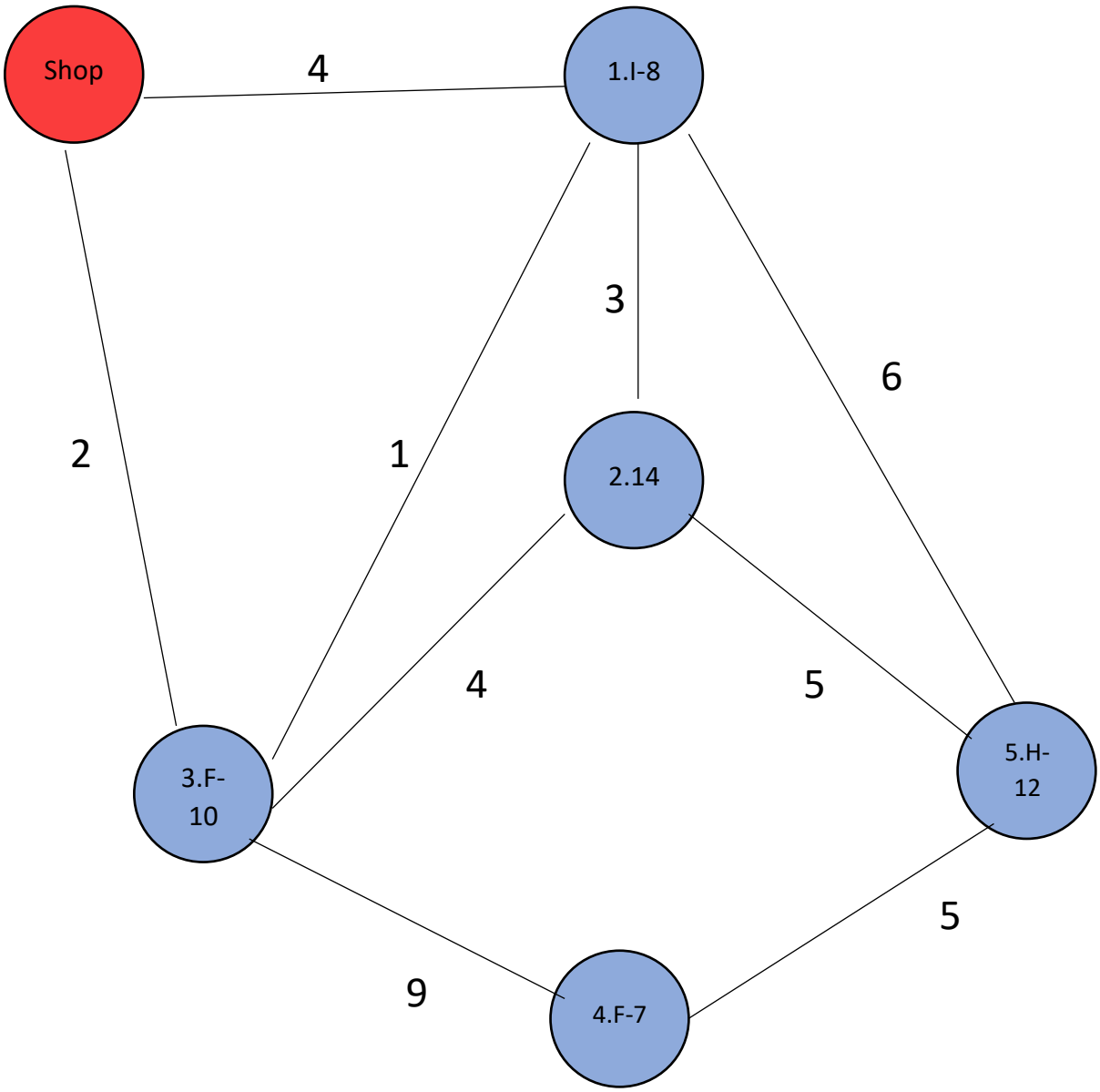
********** WareHouse **********
Location : Comsats University Islamabad, Chak Shahzad

(Maps)

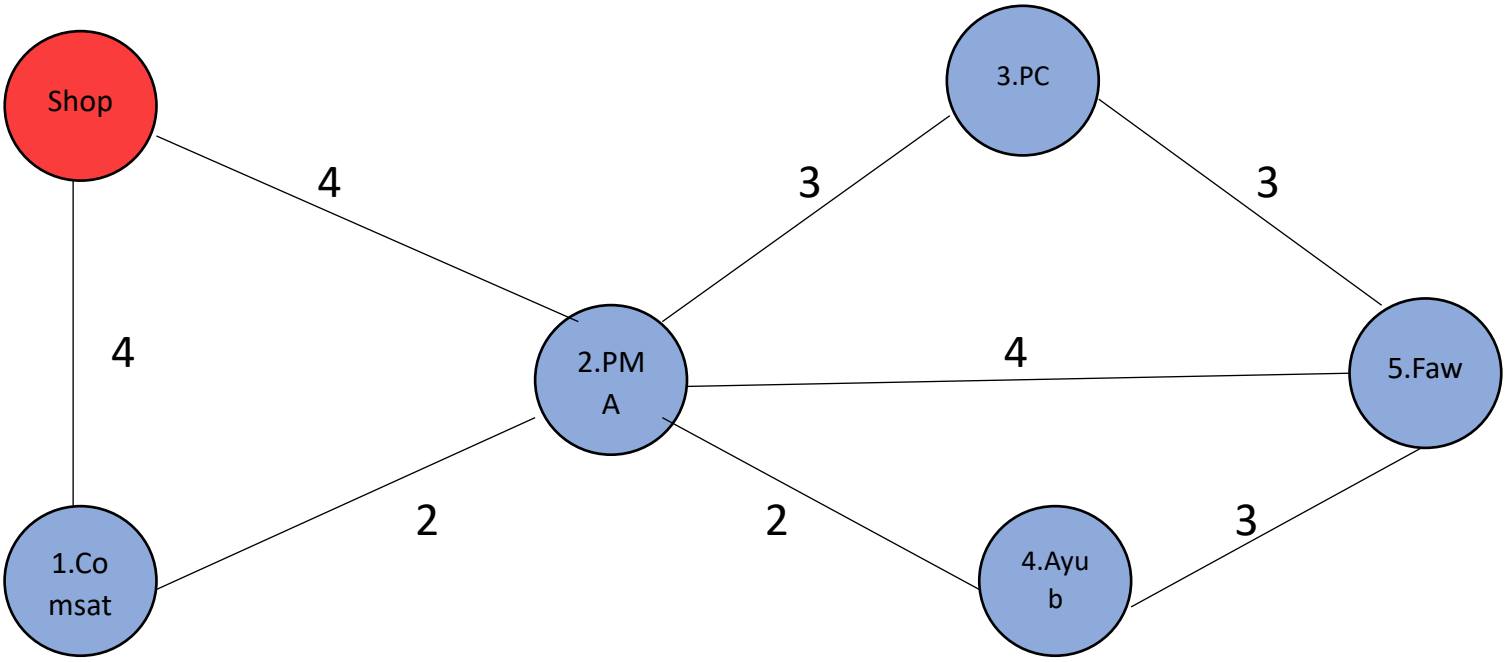
Pakistan :


1.Isb
7.Atd
8.Skr
Shop
5.Pesh
3.Khr
10.LB
2.Lhr
4.Multan
9.Gaw
6.Qut
6
5
4
17
3
16
2
10
7
3
4
8
1
11
9
5
7
13
6

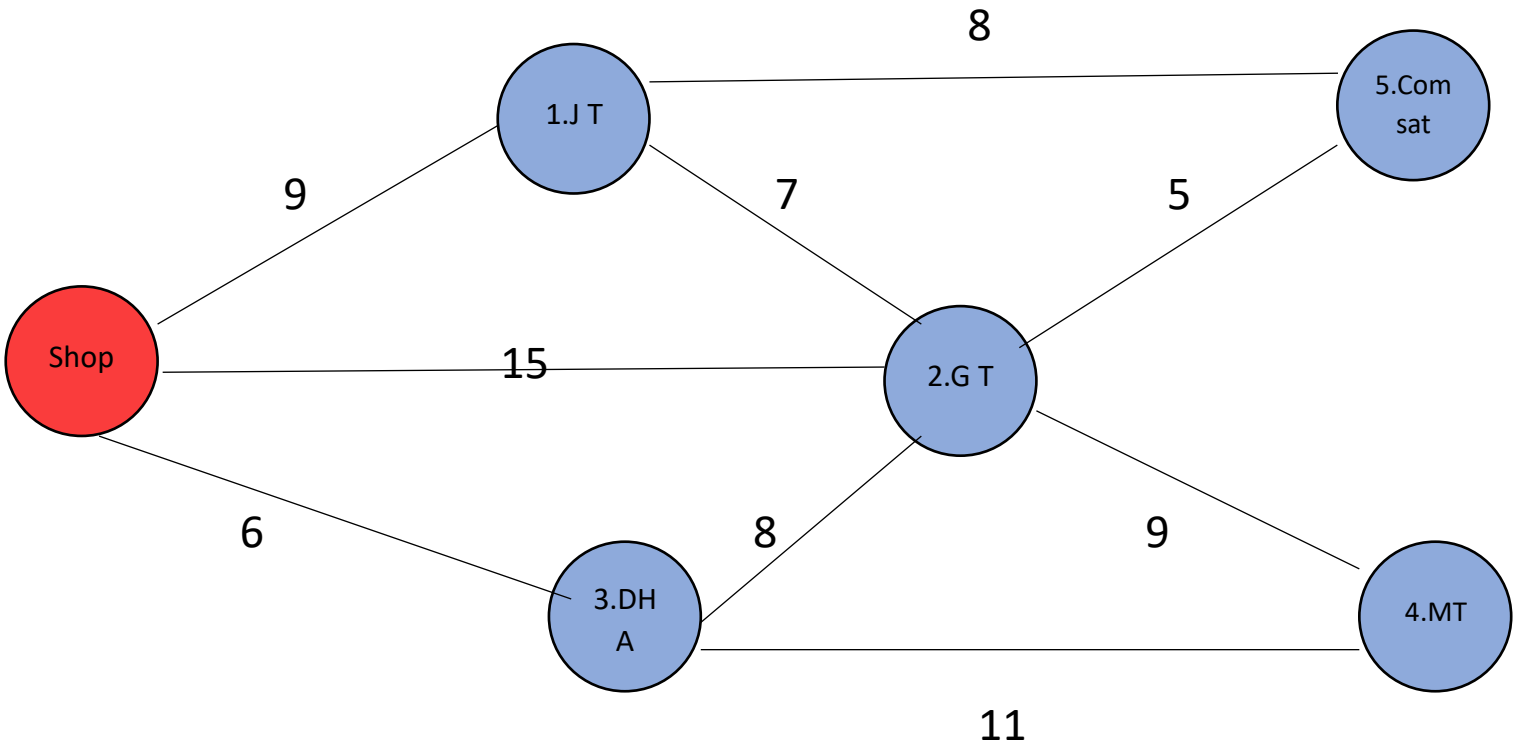
Islamabad:


Shop
4
1.I-8
2
1
3
6
2.14
4
5
3.F-10
5.H-12
9
5
4.F-7

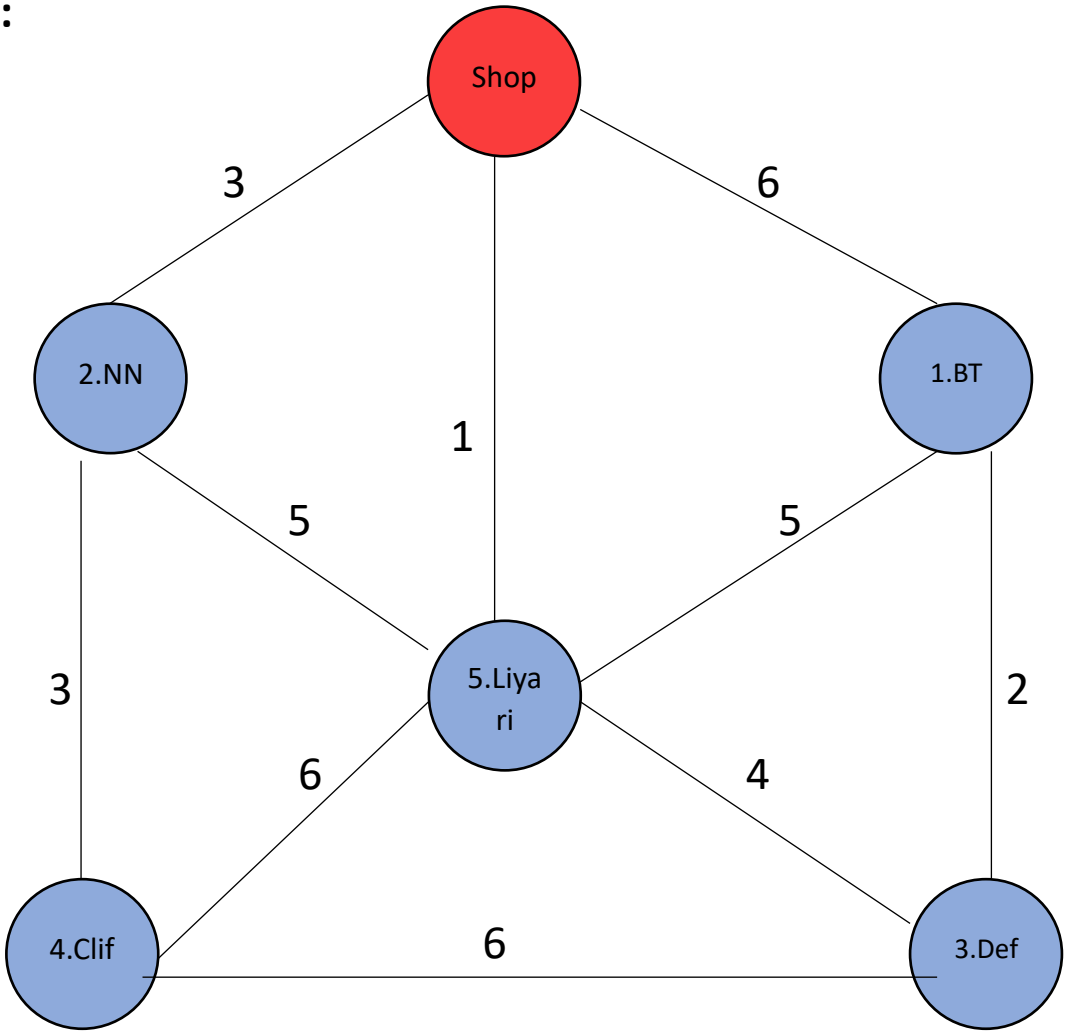
Abbottabad:



Lahore:



Karachi:



Other Cities:

- Multan
- Quetta
- Peshawar
- Skardu
- Gwadar
- Liaquat Bhag

(Adjacency matrix)

City:

```
int graph[V][V]={   {0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 3},   //Warehouse
                    {4, 0, 0, 0, 0, 0, 0, 6, 0, 0, 2},   //Islamabad
                    {0, 0, 0, 0, 9, 4, 0, 0, 0, 0, 11},  //Lahore
                    {0, 0, 0, 0, 1, 0, 0, 0, 16, 5, 0},  //Karachi
                    {0, 0, 9, 1, 0, 8, 13, 0, 10, 6, 0}, //Multan
                    {0, 0, 4, 0, 8, 0, 0, 3, 17, 0, 7},  //Peshawar
                    {0, 0, 0, 0, 13, 0, 0, 0, 0, 0, 7},  //Quetta
                    {0, 6, 0, 0, 0, 3, 0, 0, 5, 0, 0},   //Abbotabad
                    {0, 0, 0, 16, 10, 17, 0, 5, 0, 0, 0},//Skardu
                    {0, 0, 0, 5, 6, 0, 0, 0, 0, 0, 0},   //Gawadar
                    {3, 2, 11, 0, 0, 7, 7, 0, 0, 0, 0}  };//Liaquat Bhag
```

CITY CODE	City
1	Islamabad
2	Lahore
3	Karachi
4	Multan
5	Peshawar
6	Quetta
7	Abbotabad
8	Skardu
9	Gawadar
10	Liaquat Bhag

Islamabad:

```
Islamabad-----------------------
 { 0, 4, 0, 2, 0, 0 },
 { 4, 0, 3, 1, 0, 6 },    //I-8
 { 0, 3, 0, 4, 0, 7 },    //Askari
 { 2, 1, 4, 0, 9, 0 },    //F-10
 { 0, 0, 0, 9, 0, 5 },    //F-7
 { 0, 6, 7, 0, 5, 0 } }; //H-12
```

CITY CODE	AREA
1	I-8
2	Askari-14
3	F-10
4	F-7
5	H-12

Lahore:

```
--Lahore-----------------------------------
 { 0, 4, 0, 4, 0, 0 },
 { 4, 0, 0, 2, 0, 0 },    //Johar Town
 { 0, 0, 0, 3, 0, 3 },    //Garden Town
 { 4, 2, 3, 0, 2, 4 },    //DHA
 { 0, 0, 0, 2, 0, 3 },    //Model Town
 { 0, 0, 3, 0, 3, 0 } }; //Comsats Lahore
```

CITY CODE	AREA
1	Johar Town
2	Garden Town
3	DHA
4	Model Town
5	Comsats LHR

Karachi:

```
Karachi-----------------------------------
{ 0, 6, 3, 0, 0, 1 },
{ 6, 0, 0, 2, 0, 5 },    //Bharia Town
{ 3, 0, 0, 0, 3, 5 },    //North Nazimabad
{ 0, 2, 0, 0, 6, 4 },    //Defence
{ 0, 0, 3, 6, 0, 6 },    //Clifton
{ 1, 5, 5, 4, 6, 0 } }; //Liyari
```

CITY CODE	AREA
1	Bharia Town
2	North Nazimabad
3	Defence
4	Clifton
5	Liyari

Abbottabad:

```
-Abbotabad-------------------------
 { 0, 4, 4, 0, 0, 0 },
 { 4, 0, 2, 0, 0, 0 },    //comsats
 { 4, 2, 0, 3, 2, 4 },    //PMA
 { 0, 0, 3, 0, 0, 3 },    //PC
 { 0, 0, 2, 0, 0, 3 },    //Ayub
 { 0, 0, 4, 3, 3, 0 } }; //Fawara
```

CITY CODE	AREA
1	Comsats Atd
2	PMA
3	PC-Hotel
4	Ayub Medical
5	Fawara Chowk