

Experiment No: 01

Experiment Name: Finding the Shortest Path using Dijkstra Algorithm

Code:

```
# Constants
```

```
x = 3
```

```
MAX = float('inf') # Use a large number to represent infinity
```

```
def mindistance(dist, travel):
```

```
    min_value = MAX
```

```
    min_index = -1
```

```
    for i in range(x):
```

```
        if not travel[i] and dist[i] < min_value:
```

```
            min_value = dist[i]
```

```
            min_index = i
```

```
    return min_index
```

```
def print_solution(dist):
```

```
    print("Vertex \t\t Distance from Source")
```

```
    for i in range(x):
```

```
        print(f"{i} \t\t {dist[i]}")
```

```
def greedy(graph, start):
```

```
    dist = [MAX] * x
```

```
    travel = [False] * x
```

```
    dist[start] = 0
```

```
    for count in range(x - 1):
```

```
        u = mindistance(dist, travel)
```

```
        travel[u] = True
```

```

    for v in range(x):
        # Update the distance if the edge exists and the new distance is smaller
        if not travel[v] and graph[u][v] and dist[u] != MAX and dist[u] + graph[u][v] < dist[v]:
            dist[v] = dist[u] + graph[u][v]

    print_solution(dist)

def main():
    """Main function to set up the graph and execute the greedy algorithm."""
    graph = [
        [0, 2, 1],
        [2, 0, 5],
        [1, 5, 0]
    ]
    greedy(graph, 0)

if __name__ == "__main__": # Corrected condition
    main()

```

Output:

Vertex	Distance from source
0	0
1	2
2	1