Seat No.: T191094350 Name : Aryan Sirdesai

Problem Statement : Implement Greedy search algorithm for : II. Single-Source Shortest Path Problem

import sys

def greedy_search(graph, source): distances = {node: sys.maxsize for node in graph} distances[source] = 0 unvisited = set(graph.keys())

```
    while unvisited:
        current_node = min(unvisited, key=lambda node:
    distances[node])
        unvisited.remove(current_node)
        for neighbor, weight in graph[current_node].items():
            if neighbor in unvisited:
                new_distance = distances[current_node] + weight
                if new_distance < distances[neighbor]:
                    distances[neighbor] = new_distance

    return distances
```

graph = {} n = int(input("Enter the Number of Edges: ")) for i in range(n): edge = input("Enter the Edge (Source Destination Weight): ").split() source, destination, weight = edge[0], edge[1], int(edge[2]) if source not in graph: graph[source] = {} graph[source] [destination] = weight

source = input("Enter the Source Node: ") distances = greedy_search(graph, source) print(distances)

In [ ]: