Caminos mínimos

```
import sys
import heapq

def dijkstra(graph, initial_vertex, last_vertex):
    distances = {vertex: float('inf') for vertex in graph}
    distances[initial_vertex] = 0
    pq = [(0, initial_vertex)]
    heapq.heappush(pq, (0, initial_vertex))

    while pq:
        current_distance, current_vertex = heapq.heappop(pq)

        # Si llegamos al destino, devolvemos la distancia
        if current_vertex == last_vertex:
            return current_distance

        # Si la distancia es mayor que la antes registrada, ignoramos esta ruta
        if current_distance > distances[current_vertex]:
            continue

        # Analizar los vecinos con la latencia
        for neighbor, weight in graph[current_vertex]:
            new_distance = current_distance + weight

            if new_distance < distances[neighbor]:
                distances[neighbor] = new_distance
                heapq.heappush(pq, (new_distance, neighbor))

    return "unreachable"

def main():
    input_data = sys.stdin.read().strip().splitlines()
    index = 0
    test_cases = int(input_data[index])
    index += 1
    results = []

    for case_number in range(1, test_cases + 1):
        n, m, S, T = map(int, input_data[index].split())
        index += 1

        graph = {i: [] for i in range(n)}
        for _ in range(m):
            u, v, w = map(int, input_data[index].split())
```

```
            index += 1
            graph[u].append((v, w))
            graph[v].append((u, w))

        shortest_time = dijkstra(graph, S, T)

        results.append(f"Case #{case_number}: {shortest_time}")

    print("\n".join(results))

main()
```

My Submissions

#		Problem	Verdict	Language	Run Time	Submission Date
29931282	10986	Sending email	Accepted	PYTH3	1.580	2024-11-02 00:16:43