Harshit Hiremath

1BM18CS036

07/12/20

CN Lab8

Implement Dijkstra's Algorithm to compute shortest path through a weighted undirected graph

```
def dijkstra(graph, src):
    distance = [infinite]*n
    distance [src] = 0
    final_selected = [(src, distance[src])]
    curr_vertex = src
    while len(final_selected) <n:
        min_vertex, min_dist= -1, infinite
        for neighbour in graph[curr_vertex]:
            vertex, weight =neighbour
            distance [vertex]= min (
                        distance [curr_vertex]+weight,
                        distance [vertex])

        for vertex in range(n):
            if distance [vertex] <=min_dist and (vertex,
                                        distance[vertex])
                      not in final_selected:
                min_vertex, min_dist = vertex, distance [vertex]

        final_selected.append((min_vertex, min_dist))
        curr_vertex = min_vertex
    print('Vertex\t Distance')
    for v, d in final_selected:
        printf(f'{v}\t{d}')
```

(for infinite, we use math library's inf constant, i.e. math.inf)