7-12-20

CN_lab

Dijkstras algorithm

B Praneeth
1BM18CS023

```
def dijkstrasAlgorithm(start, edges):
    int = float('inf')                    # edges → Adjacency matrix.
    visited = []
    n = len(edges)
    array = [inf] * n
    path = [[None]] * n
    array[start] = 0

    curr_vertex = start
    path[start] = [start]
    print(path)

    while (len(visited) < n):
        smallest = inf
        for k in range(n):
            if (array[k] < smallest and k not in visited):
                smallest_i = k
                smallest = array[k]
        copy(path[curr_vertex], path[smallest_i])
        path[smallest_i].append(smallest_i)
        current_vertex = smallest_i
        current_vertex_dist = array[curr_vertex]
```

```
    for edge in edges[curr_vertex]:
        dest_vertex = edge[0]
        dist = curr_vertex_dist + edge[1]
        if(dist < array[dest_vertex]):
            array[dest_vertex] = dist

    visited.append(curr_vertex)

for i in range(len(array)):
    if(array[i] == inf):
        array[i] = -1

return array, path

dijkstraSAlgorithm(0, [[1,7],[2,6]],[3,20]])
```