



Islamic University of Technology

CSE 4810

Algorithm Engineering Lab

Lab 4

Tasnimul Hasnat

190041113

CSE 1A

April 17, 2024

Two tasks were given, solved them in Leetcode.

Task 1

```
class Solution:
    def findTheCity(self, N, edges, distanceThreshold):
        Graph = defaultdict(list)
        for u, v, w in edges:
            Graph[u].append([v, w])
            Graph[v].append([u, w])

        def dijkstra(i):
            pq = []
            heappush(pq, [0, i])
            path = [inf] * N
            path[i] = 0

            while pq:
                dist, node = heappop(pq)
                if dist > distanceThreshold:
                    break
                for nei, w in Graph[node]:
                    if dist + w < path[nei]:
                        path[nei] = dist + w
                        heappush(pq, [path[nei], nei])

            cnt = -1
            for i in range(N):
                cnt += path[i] <= distanceThreshold
            return cnt

        cnts = {}
        mini, maxi = N + 1, 0
        for i in range(N):
            cnts[i] = dijkstra(i)
            mini = min(cnts[i], mini)
            if cnts[i] == mini:
                maxi = max(maxi, i)
        return maxi
```

Task 2

```
class Solution:
    def minCostConnectPoints(self, points):
        distances = [[0, 0]] + [[float('inf'), i] for i in range(1, len(points))]
        result = 0
        while distances:
            closest = min(distances)
            result += closest[0]
            x, y = points[closest[1]]

            closest[:] = distances[-1][:]
            distances.pop()

            for dist in distances:
                x2, y2 = points[dist[1]]
                dist[0] = min(dist[0], abs(x - x2) + abs(y - y2))
        return result
```