CSE 106: Offline Assignment

Hierarchical Emergency Network Optimization

(Using Divide & Conquer and Greedy Algorithms)

Bangladesh University of Engineering and Technology, BUET
Department of CSE

1. Introduction

This assignment focuses on designing an efficient large-scale infrastructure network using two fundamental algorithm design paradigms:

- Divide and Conquer
- Greedy Algorithms

Students must combine both paradigms into a hybrid algorithm under structural and optimization constraints.

2. Problem Description

A country is building an emergency communication network among N cities. Each city has geographic coordinates and can be connected via communication links.

The cost of connecting city i and j is:

$$Cost(i, j) = \alpha \cdot Distance(i, j) + \beta \cdot Risk(i, j)$$

Where:

$$Distance(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

If no risk value is provided for a pair, then $Risk(i, j) = 0$.

Global Constraints

- The entire network must be connected.
- Each city can have at most K direct connections.
- Each regional cluster must contain at least one cycle.
- $10^4 \leq N \leq 10^5$
- $2 \leq K \leq 10$

Your algorithm must run in strictly better than $O(N^2)$ time.

3. Input Specification and Parameter Explanation

The general input format is:

N K M α β

```
Cities:
<ID> <x> <y>
...
Risk:
<u> <v> <risk_value>
...
```

3.1 Parameter Explanation

1. N **— Number of Cities**

- Total number of cities.
- Cities are labeled from 1 to N.
- Determines problem size and efficiency requirements.

2. K **— Maximum Degree Constraint**

- Maximum number of direct connections allowed per city.
- If degree of any city exceeds K, solution is invalid.
- Prevents trivial star-shaped greedy solutions.

3. M **— Maximum Cluster Size**

- Divide and Conquer stopping threshold.
- If cluster size $\leq M$, solve locally using greedy.
- Controls recursion depth and merge complexity.

4. α **— Distance Weight**

- Controls importance of geographic distance.
- Larger α prioritizes shorter edges.

5. β — Risk Weight

- Controls importance of communication risk.
- Larger β prioritizes safer (lower risk) edges.

6. Cities Section

Each city line contains:

- City ID
- x coordinate
- y coordinate

Coordinates are used for:

- Distance calculation
- Spatial partitioning

7. Risk Section

Each line:

```
u v risk_value
```

- Represents additional risk between cities u and v.
- Only listed pairs have non-zero risk.
- All unspecified pairs have risk $= 0$.

This ensures sparse representation and prevents $O(N^2)$ storage.

4. Algorithmic Requirements

4.1 Phase 1: Divide and Conquer

- Recursively partition cities by median x-coordinate.
- Stop when cluster size $\leq M$.
- Solve locally within clusters.
- Merge clusters carefully.

4.2 Phase 2: Greedy Optimization

- Select minimum-cost edges.
- Respect degree constraint K.
- Add minimum-cost redundancy edges.

You must clearly explain:

- Recurrence relation
- Time complexity
- Greedy choice justification
- Feasibility of merge step

5. Output Format

```
Total Cost: <value>
Edges:
u1 v1
u2 v2
...
```

Order does not matter, but all constraints must hold.

6. Sample Input and Output

6.1 Sample 1: Basic Connected Case

Input

```
8 3 3 1 2
Cities:
1 1 1
2 2 2
3 3 1
4 8 1
5 9 2
6 8 3
7 4 8
```

```
8 5 9

Risk:
1 2 1
2 3 2
4 5 1
7 8 1
```

Output (One Valid Solution)

```
Total Cost: 28.47
Edges:
1 2
2 3
3 7
7 8
4 5
5 6
3 4
2 1
5 4
```

—

6.2 Sample 2: Degree Constraint Tight Case

Input

```
6 2 3 1 1
Cities:
1 0 0
2 1 0
3 2 0
4 3 0
5 4 0
6 5 0

Risk:
```

Explanation

Since $K = 2$, each city can connect to at most two others.

Output (One Valid Solution)

```
Total Cost: 5.00
Edges:
1 2
2 3
3 4
4 5
5 6
```

(Note: No redundancy possible without violating degree constraint.)

—

6.3 Sample 3

Input

```
4 3 4 1 0
Cities:
1 0 0
2 0 1
3 1 1
4 1 0

Risk:
```

Output (One Valid Solution)

```
Total Cost: 4.00
Edges:
1 2
2 3
3 4
4 1
```

—

6.4 Sample 4

Input

```
5 3 2 1 10
Cities:
1 0 0
2 1 0
```

```
3 2 0
4 10 0
5 11 0

Risk:
1 2 5
2 3 5
```

Explanation

Because $\beta = 10$, risk heavily influences cost. Algorithm must avoid high-risk edges even if distance is small.

Output (One Valid Solution)

```
Total Cost: 21.00
Edges:
1 3
3 4
4 5
2 3
```

—

6.5 Sample 5

Input

```
6 3 2 1 0
Cities:
1 0 0
2 1 0
3 2 0
4 100 0
5 101 0
6 102 0

Risk:
```

Explanation

Divide step splits into two clusters: {1,2,3} and {4,5,6}.

Correct merge must connect closest boundary nodes.

Output (One Valid Solution)

```
Total Cost: 102.00
Edges:
1 2
2 3
4 5
5 6
3 4
```

—

6.6 Sample 6

Input

```
7 3 3 2 5
Cities:
1 0 0
2 2 0
3 4 0
4 6 0
5 8 0
6 10 0
7 12 0

Risk:
3 5 1
2 6 2
```

Explanation

Sparse risk edges change greedy edge ordering. Students must incorporate risk properly into cost calculation.

Output (One Valid Solution)

```
Total Cost: 48.00
Edges:
1 2
2 3
3 4
4 5
5 6
6 7
3 5
```

—

Important Notes About Sample Outputs

- Outputs shown are only **one valid solution**.
- Edge order does not matter.
- Multiple valid solutions may exist.
- All constraints must be satisfied:
  - Connectivity
  - Degree $\leq K$
  - At least one cycle per cluster

7. Evaluation Criteria

Criteria	**Marks**
Correct use of Divide & Conquer	25
Correct use of Greedy	25
Correctness Proof	15
Time Complexity Analysis	15
Handling Edge Cases	10
Code Efficiency	10
Total	**100**

No marks will be provided for solutions that do not clearly justify algorithmic design decisions.