CS206 - Algorithms & Complexity

Assignment 1 Report

Student: Elind Kusha

Problem Description

BigWeather uses a distributed system of computing nodes called dynos to perform weather forecasting. Each dyno requires access to a high-performance cache component called a bucket. A dyno can either:

1. Host a bucket itself (at a fixed bucket cost), or

2. Connect to another dyno via a bond (at a fixed bond cost) to access a bucket hosted elsewhere.

Given:

- The number of dynos `n`

- Number of possible bonds `k`

- Cost to host a bucket

- Cost to create a bond

And a list of valid bonds, the task is to find the minimum total cost to ensure that every dyno can access a bucket, either directly or via connected dynos.

Data Structures Used

- Graph representation: Dynos and the hosting option are modeled as nodes and edges.

- Edge list: To represent all bonds and hosting options.

- Union-Find (Disjoint Set Union): To efficiently manage connectivity and detect cycles during MST construction.

Algorithm Description (Kruskal's Algorithm)

We approach this problem using a modified version of Kruskal's Minimum Spanning Tree (MST) algorithm with a virtual node:

Pseudocode:

1. Read input and initialize an edge list

2. Add an edge from a virtual node 0 to each dyno node i with cost = bucket_cost

3. Add each bond as an undirected edge between two dynos with cost = bond_cost

4. Sort all edges by cost

5. Initialize Union-Find for all dynos including the virtual node

6. Iterate over the sorted edges:

- If two nodes are not connected, union them and add the cost

7. Continue until all dynos are connected to at least one bucket

8. Output the total cost

Correctness

The correctness of this approach is guaranteed by Kruskal's algorithm:

- It always returns the minimum spanning forest (tree for connected components)

- By connecting each dyno to the virtual node (representing hosting), we ensure that each dyno is either hosting a bucket or connected to someone who is.

Time Complexity

- Sorting edges: O(E log E), where E is the number of edges (bonds + n bucket edges)

- Union-Find operations: Nearly O(1) per operation (with path compression)

- Overall Complexity: O(E log E) which is efficient for this problem scale.

Example Input

6 5 4 3

1 2

1 3

4 5

6 4

2 3

Sample Output

20

Conclusion

The problem was successfully modeled as a graph with an artificial node and solved using a classic MST approach. The implementation ensures minimal cost while satisfying all connectivity constraints.

Bonus Consideration (Optional)

- Count number of minimum cost configurations (not implemented)

- Visualize one minimum cost configuration (can be added upon request)

Prepared by

Elind Kusha