Project 3 Written Report

CPSC 335 - Algorithm Engineering

Spring 2025

Names:

Anthony Le | antlecsuf@csu.fullerton.edu

Ryan Nishikawa | ryannishikawa48@csu.fullerton.edu

Dylan Tran | dylanht341@csu.fullerton.edu

Jasmine Youssef | JasmineYoussef@csu.fullerton.edu

Algorithm 2: Delivery Route Planning

Problem Description

You are given a set of delivery routes between distribution centers, each with a cost. The objective is to determine the cheapest route from a starting distribution center (src) to a destination center (dst) with at most k stopovers (intermediate transfers). Return the minimum delivery cost or -1 if no valid route exists.

Sample Input and Output

Example 1:

routes = [

[0, 1, 100],

[1, 2, 100],

[0, 2, 500]

]

src = 0

dst = 2

k = 1

Output: 200

Explanation: 0 -> 1 -> 2 costs 100 + 100 = 200

Example 2:

routes = [

[0, 1, 100],

[1, 2, 100],

[0, 2, 500]

]
src = 0

dst = 2

k = 0

Output: 500

Explanation: Only direct path 0 -> 2 is valid

Example 3:

routes = [

[0, 1, 100],

[1, 2, 100],

[0, 2, 500]

]

src = 0

dst = 3

k = 1

Output: -1

Explanation: No valid route to center 3

Pseudocode:

```
START
    INPUT routes, src, dst, k
    CREATE graph from routes
    INITIALIZE minHeap with (cost=0, node=src, stops=0)

    WHILE minHeap is not empty:
        POP (cost, current_node, stops) from minHeap
        IF current_node == dst:
            RETURN cost
        IF stops > k:
            CONTINUE
        FOR each neighbor of current_node in graph:

            PUSH (cost + edge_cost, neighbor, stops + 1) to minHeap

    RETURN -1
END
```

Mathematical Analysis and Efficiency

- **Time Complexity**:
- Let n be the number of centers and e be the number of routes.
- Graph construction: O(e)
- Min-heap operations: O(k * e * log n)

Space Complexity: O(n + e) for graph, O(n * k) for min-heap.

Efficiency Class: O(k * e * log n), efficient for sparse graphs and limited k.