COMP3010 Algorithm Theory Assignment 1

Chaz Lambrechtsen

45426317

1 Contents

2 Algorithm Explanation

2.1 Overview

In order to achieve the minimum cost and maximum flow of a network: G = (v, e) is a maximum flow (of packages in this problem) with the lowest possible cost. This problem combines maximum flow (getting as much flow as possible from the warehouse to the disaster zone) with shortest path (reaching from the warehouse to the disaster zone with minimum cost).

2.2 Ford Fulkerson Maximum Flow

First, we set the flow of each edge to zero. Then we look for an augmenting path from Warehouse to Danger zone. An augmenting path is simple path in the residual graph, i.e. along the edges whose residual capacity is positive. If the path is found, then we can add increase the flow along the edges. We keep on searching for augmenting paths and increasing the flow. When there is no longer an augmenting path, the flow is maximal.

2.3 Bellman-Ford Minimum Cost

The Bellman–Ford algorithm is an algorithm that computes shortest paths from the warehouse vertex to the disaster zone vertex in the weighted digraph matrix represented as cost[ ][ ]. The logic of Bellman-Ford in this problem is as follows:

- Store the capacity of an edge and the cost of that edge in separate 2D arrays.
- If a flow exists, calculate the distance, value = distance + edge – edge [i] – cost[i].
- Compare the distance values in distance[ ] with value and keep updating until minimum flow is obtained.

3 Algorithm Implementation

3.1 Flow Network

A 2D array is implemented in the code to represent the flow network matrix. This is implemented as follows:

int flow[ ][ ];

For example, a flow matrix for final test p3_2.in

Flow network:

```
0 15 5 0
0 0  5 10
0 0  0 10
0 0  0 0
```

3.2 Residual Network

The residual network is represented in two 2D arrays, cost and capacity which is implemented as the following:

int cost[ ][ ];
int capacity[ ][ ];

Cost is a matrix in which cost[a][b] is the cost of sending one package of flow along a directed edge from vertex a to vertex b. Similarly, Capacity is a matrix in which there is a defined capacity for edges from vertex a to b.

4 Structure

Debug / testing method to print a given matrix

```
void printMatrix(int[][] a)
```

MaximumFlow function utilises the pathExists helper function.

```
int[] maximumFlow(int warehouseNode, int disasterNode)
```

Helper function for finding an existing path

```
boolean pathExists(int warehouseNode, int disasterNode)
```