OU3 - Mandatory Exercise 3

ID1018

October 22, 2014

Algorithms

The shortest path

A problem: find intermediate stations on the shortest path

A commuter traffic system has four zones: Z_1, Z_2, Z_3 and Z_4. In zone Z_1 there is only one station, X, and zone Z_4 only contains station Y. In zone Z_2 there are stations $U_1, U_2, \ldots, U_m$ (m being a positive integer), and zone Z_3 contains the stations $V_1, V_2, \ldots, V_n$ (n is a positive integer).

There are direct paths between station X and all stations in zone Z_2. The zones Z_2 and Z_3 are *well connected* to each other; there is a direct path from any station in one zone to any station in the other zone. There is also a direct path between any station in zone Z_3 and station Y. There are no other paths.

For any integer $i, 1 \leq i \leq m$ the following holds: the length of the path between station X and station U_i is a_i.

For any integer $i, 1 \leq i \leq m$, and any integer $j, 1 \leq j \leq n$, the following holds: the length of the path between station U_i and V_j is b_{ij}.

For any integer $j, 1 \leq j \leq n$, the following holds: the length of the path between station V_j and station Y is c_j.

A path between stations X and Y passes through one station in zone Z_2 and one station in zone Z_3. An intermediate station in each of the zones Z_2 and Z_3 is to be chosen, so that the path between station X and station Y is as short as possible.

It may be the case that there are several shortest paths. If so, intermediate stations on one of these paths are to be selected.

Z_1	a_i	Z_2	b_{ij}	Z_3	c_j	Z_4	length
X		U_1		V_1		Y	
X		U_1		V_2		Y	
X		U_1		V_3		Y	
X		U_1		V_4		Y	
X		U_2		V_1		Y	
X		U_2		V_2		Y	
X		U_2		V_3		Y	
X		U_2		V_4		Y	
X		U_3		V_1		Y	
X		U_3		V_2		Y	
X		U_3		V_3		Y	
X		U_3		V_4		Y	

Table 1: Template for path table

Exercises on the problem

1. Decide an instance of the problem where $m = 3$ and $n = 4$. Choose the path lengths. Specify this instance in a drawing. Indicate stations, paths and path lengths.

2. Specify the same instance in a table. The table should be laid out as in table 1.

Solve the instance of the problem using pen and paper. Examine all routes and determine the intermediate stations for the shortest path (enter the lengths in the column *length* and select the intermediate stations that correspond to the minimum length).

3. Find a memory-efficient algorithm that solves this problem in a general case — use an update strategy. Describe this algorithm in two ways: with words and with pseudocode.

The description shall be in the following form:

PROBLEM
problem description

ALGORITHM

PRECONDITIONS
specify the preconditions of the algorithm

POSTCONDITIONS
specify the postconditions of the algorithm

STEPS IN THE ALGORITHM
describe the steps of the algorithm in words

STEPS IN THE ALGORITHM — PSEUDOCODE
describe the steps of the algorithm symbolically

4. Create a Java program that can solve different instances of the problem. Use the program with two separate instances and explain the results you got.

There shall be two classes, `TheShortestPath` and `DetermineTheShortestPath`. The first class shall look like this:

```
class TheShortestPath
{
  // The method intermediateStations returns a vector of the
  // intermediate stations that are on the shortest path.
  // The ordinal number of the first station is located in
  // index 1 of the vector, and the second station on index 2.
  public static int [] intermediateStations (double [] a,
                                             double [][] b,
                                             double [] c)
  {
    // *** WRITE YOUR CODE HERE ***
  }

  // The method length returns the length of the shortest path.
  public static double length (double [] a,
                               double [][] b,
                               double [] c)
  {
    // *** WRITE YOUR CODE HERE ***
  }
}
```

The class `DetermineTheShortestPath` shall contain the main method, where data specific to the problem instance is entered, and from where methods in the class `TheShortestPath` are called.