National Taiwan University of Science and Technology
Department of Electrical Engineering

Algorithm Design and Application, Fall 2020

Programming Assignment #1

Monotonic Routing (due December 4, 2020 (Friday) on-line)

1. Problem Description

Given is an m × n grid graph, the cost of each graph edge (u, v) as $d(u, v)$, a source grid S, and a target grid T (see Figure 1(a)). Let $D(S, g)$ denotes the minimum cost of a non-detour path from S to a grid g. You are asked to write a program that finds a monotonic path from the source grid to the target grid, i.e., compute $D(S, T)$, and reports the total edge cost, the number of grids, and the grid coordinates of the path.

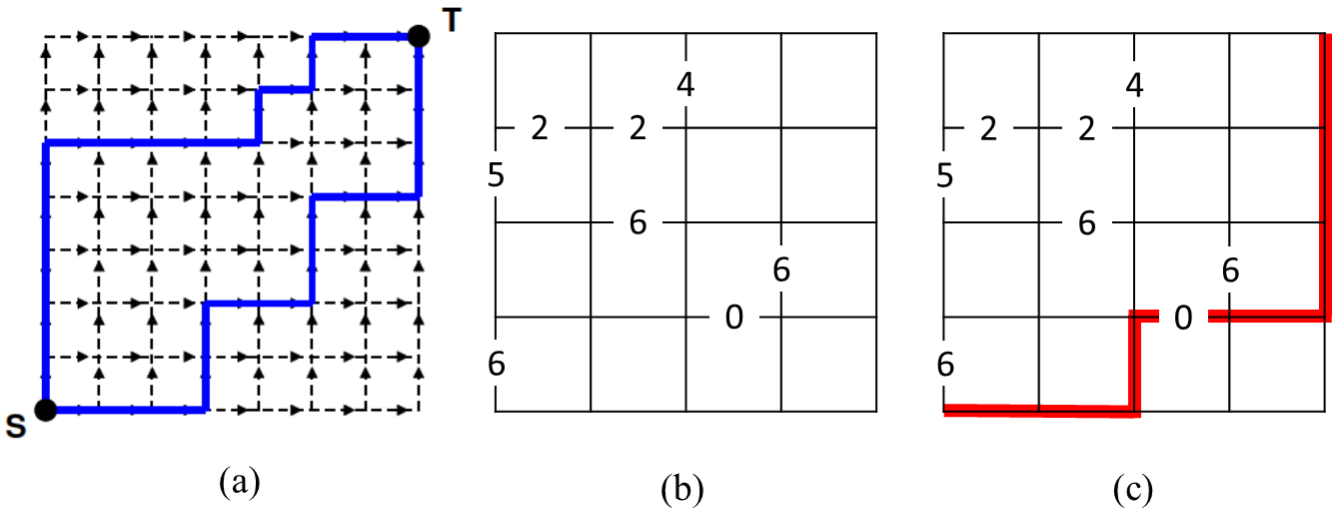

Figure 1: An example of monotonic routing.

2. Input

The file format for the monotonic routing is illustrated, with comments in italics (these will not be in actual input files). The 1st line gives the problem size in terms of the indices of the left-bottom and the right-top grids (the indices of the left-bottom grid will always be (0,0)). Each edge connecting two grids has either default cost or non-default cost. The default cost is given in the 2nd line. The 3rd line gives the number of edges that have non-default costs. Start from the 4th line, the cost difference compared to the default cost of each non-default edge is separately given. Finally, the coordinates of the source and the target grids are listed in the last two lines.

The input file format is as follows:

BoundaryIndex # # # # *//the indices of the left-bottom and the right-top grids*
DefaultCost # *//the default cost*
NumNonDefaultCost # *//the number of non-default edges*
x1 y1 x2 y2 # *// the cost difference of the non-default edge between (x1,y1) and (x2,y2)*
...
//repeat for the number of non-default edges
Source xS yS *//the coordinate of the source grid*
Target xT yT *//the coordinate of the target grid*

3. Output

The resulting monotonic path needs to be described in the output file. The 1st line gives the total edge cost in the resulting path, and the 2nd line gives the number of grids on the path. After that, the consecutive grids in the path from source to target have to be listed in order. The output file format is as follows:

RoutingCost [total edge cost]
RoutingPath [# of grids, k]
[x_1] [y_1]
[x_2] [y_2]
...
[x_k] [y_k]
//repeat for the number of grids in the path

Note that, x_1 and y_1 must be the same as xS and yS, and xk and yk must be the same as xT and yT in the input file respectively.

Here is an input/output example of Figures 1(b)/(c):

Sample Input	Sample Output
BoundaryIndex 0 0 4 4	RoutingCost 21
DefaultCost 3	RoutingPath 9
NumNonDefaultCost 8	0 0
3 1 3 2 3	1 0
2 3 2 4 1	2 0
2 1 3 1 -3	2 1
1 3 2 3 -1	3 1
1 2 2 2 3	4 1
0 3 1 3 -1	4 2
0 2 0 3 2	4 3
0 0 0 1 3	4 4
Source 0 0	
Target 4 4	

4. Language/Platform

(a) Language: C or C++.
(b) Platform: Unix/Linux or Windows.

5. Command-line Parameter

In order to test your program, you are asked to add the following command-line parameters to your program (e.g., ./MR 5x5.in 5x5.out):

[executable file name] [input file name] [output file name]

6. Submission

You need to submit the following materials in a .tar or a .zip file (e.g., m10907400-p1.zip) at the course website by the deadline: (1) source codes, (2) executable binaries, and (3) a text readme file (readme.txt) stating how to build and use your programs.

7. Grading Policy

This programming assignment will be graded based on (1) the correctness (please include all libraries and header files that are required to compile your program on Linux), (2) readme.txt, (3) solution quality, and (4) running time. Please check these items before your submission.

8. Online Resources

Sample input files (*.in), sample parser, and sample readme.txt can be found at the course website.