Homework C

U-bike Management System

Teacher: Yu-Ling Hsueh (hsueh@cs.ccu.edu.tw)
Responsible TA: Chia-Chun Lin (LCC103P@cs.ccu.edu.tw)
Release Date: 2015.12.01 > Deadline: 2015.12.17 11:59 pm

The Taipei City Government provides a convenient bike service, named YouBike, to the public. The way people use YouBike service is via renting a bike at one YouBike station and returning it at any stations.

YouBike is very popular now, so that people sometimes cannot find a bike at the YouBike stations. To solve it, The Taipei City Government may set up a YouBike Management Center (YMC) to monitor the capacity of all the YouBike stations in the future. In addition, YMC has to keep all the YouBike stations **half-full** (the most efficient condition). While a station is in **full** or **empty** status, YMC will **collect** or **send** bikes to adjust the number of bikes at each YouBike station. Meanwhile, all the stations on the **delivery route** will may be adjusted.

YMC always chooses the shortest path as the delivery route if there is only one shortest path. However, YMC will choose the path that needs **least** number of bikes to be **sent from YMC** and **taken back to YMC,** if there are two or more shortest paths.

Figure 1 shows an example. The nodes represent the YMC and YouBike stations. The number on an edge is the transmission cost between two nodes. The pair of numbers on each node shows the current capacity v.s. the maximum capacity of a YouBike station. To solve this example, there are two shortest paths, given YMC as a source node and $Station_3$ as a sink node.

$Path_1$: YMC -> $Station_1$ -> $Station_3$
In this case, a bike carrier has to send 4 bikes from YMC, because the bike carrier can collect one bike from $Station_1$, so that 5 bikes can be delivered to $Station_3$.

$Path_2$: YMC > $Station_2$ > $Station_3$
In this case, a bike carrier has to send 3 bikes from YMC, because the bike carrier can collect the rest two bikes from $Station_2$, so that 3 bikes can be delivered to $Station_3$.

We have described that YMC will choose the path that needs least number of bikes to be delivered from YMC and taken back to YMC. Therefore, Path 2 will be choosen as the delivery route. Furthermore, no matter YMC choose which path as the delivery route, the bike carrier doesn't need to take any bikes back to YMC in $example_1$.

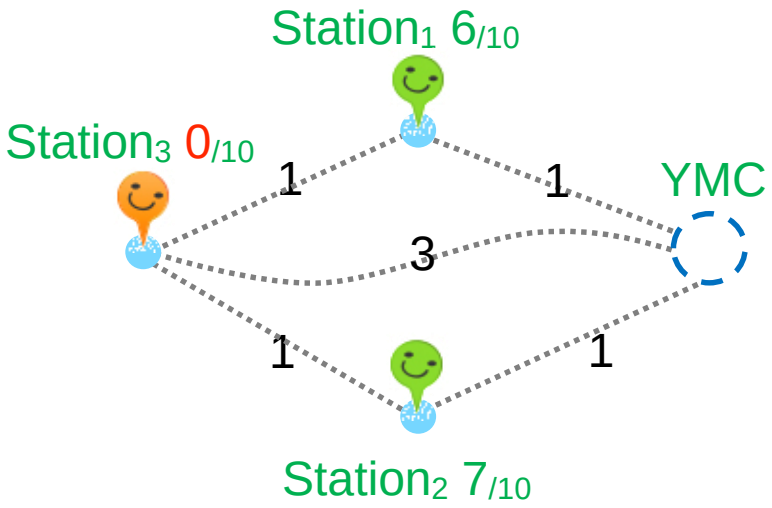

Figure 1. $Example_1$

Let's have example$_2$ shown in Figure 2. To solve this example, there are two shortest paths, given YMC as a source node and Station$_3$ as a sink node.

Path$_1$: YMC -> Station$_1$ -> Station$_3$
In this case, a bike carrier has not to send any bikes from YMC. However, the bike carrier has to take 8 bikes back from Station$_1$ to YMC.

Path$_2$: YMC > Station$_2$ > Station$_3$
In this case, a bike carrier has not to send any bikes from YMC. However, the bike carrier has to take 7 bikes back from Station$_1$ to YMC.

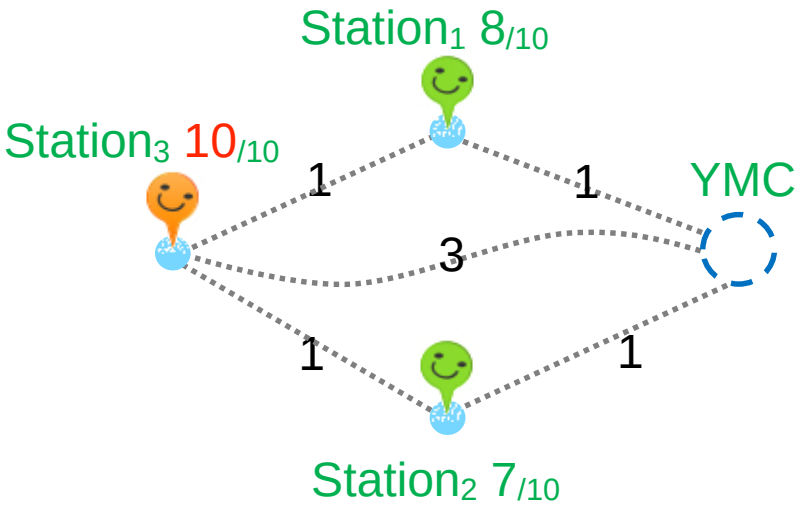

Figure 2. Example$_2$

We have described that YMC will choose the path that needs least number of bikes to be delivered from YMC and taken back to YMC. Therefore, Path 2 will be choosen as the delivery route in example$_2$.

Input Specification

Each input is a test case. For each case, the 4 numbers in first line are:

1. the maximum capacity of a YouBike station
2. the number of the YouBike stations (In example$_1$, the value is 3)
3. the destination YouBike station
4. the number of road segments (in example$_1$, this value is 5)

The numbers in second line represent the current number of bikes at each YouBike stations.

Subsequently, an adjacency list is given to represent the graph. There are 3 numbers in each line, where the first and the second numbers are two adjacent YouBike stations, and the third number is the transmission cost between them. Remember that the ID for YMC is always **0**. All the numbers in a line are separated by a space.

Output Specification

You have to print out the result in one line and follow the given specification. First number in an output is the number of bikes that must be sent from YMC. Then after one space, print the delivery route with the format: 0>station id$_x$>...->station id$_y$. Finally after another space, print the number of bikes that the bike carrier has to take back to YMC. Now, we can have sample input and output for example$_1$ and example$_2$.

Sample$_1$ Input:

```
10 3 3 5
6 7 0
0 1 1
0 2 1
0 3 3
1 3 1
2 3 1
```

Sample$_1$ output:

```
3 0>2>3 0
```

Sample$_2$ Input:

```
10 3 3 5
8 7 10
0 1 1
0 2 1
0 3 3
1 3 1
2 3 1
```

Sample$_2$ Output

```
0 0>2>3 7
```

Readme, comments and style

An indicator for good source code is readability. To keep source code maintainable and readable, you should add comments to your source code where reasonable. A consistent coding style also help a lot in reading source code. For this assignment, please also compose a small readme-file in *.txt format and name it \README.TXT". This file should contain a brief explanation of how to use your program. Please remember to have your source code comments and readme file in English.

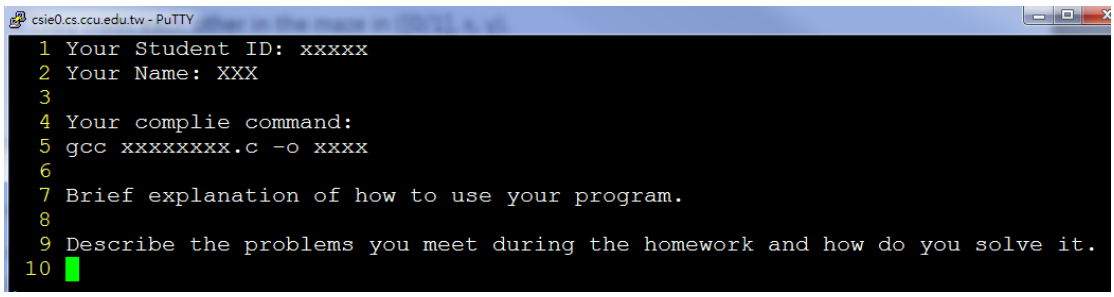

Figure 3. Readme file format

Grading (for TA)

The TA(s) will mark and give points according to the following grading polices:

90% Run your program with 5 testing data on ***PC^2***.

- 15% Testing $Data_1$
- 15% Testing $Data_2$
- 15% Testing $Data_3$
- 15% Testing $Data_4$
- 30% Advanced Testing Data

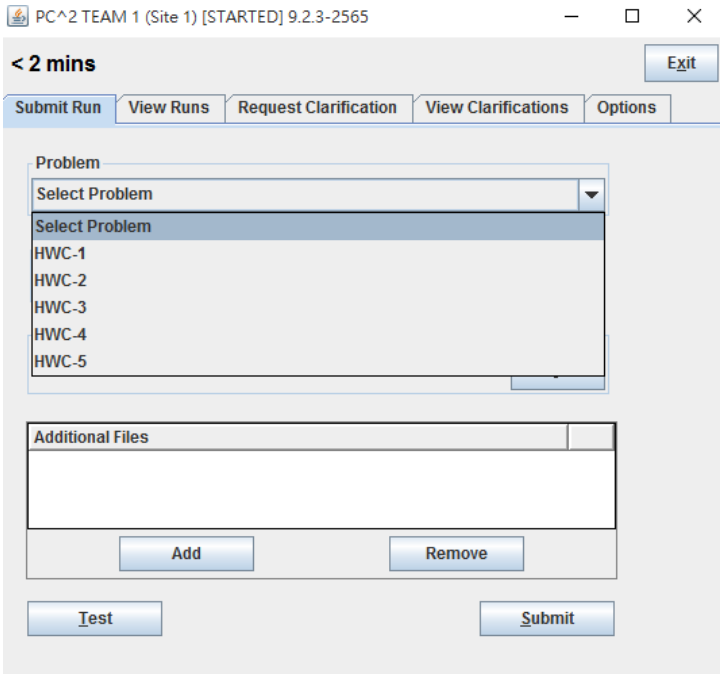

Figure 4. Screenshot of *PC^2*

10% Readme, comments and style.

Provide a README.TXT that contains information about the program. Source code is readable and has comments where reasonable.

How to submit

PuTTY and PieTTY are free and open-source terminal emulator, serial console and network file transfer application. These two tools can help you to easily access the workstation of CSIE.

Host name: csie0.cs.ccu.edu.tw

To make sure that your program can run on the workstation, you MUST USE a Makefile to compile your program. If Makefile is not submitted or do not work well, you will not get the scores of HWC.

The files you MUST submit:

1. Source Code
2. README.TXT
3. Makefile

To submit your file electronically, enter the following command in csie workstation:

trunin ds.HWC [your _les...]

To check the files you had turnin, enter the following command in csie workstation:

trunin -ls ds.HWC

You can see other description about turnin from following link:
https://www.cs.ccu.edu.tw/lab401/doku.php?id=turninhowto