CMPE 343
Fall 2020
Programming Homework #4

This assignment is due by 23:55, 20 December 2020.

Late submissions: 20 points will be deducted for each late day.

PART I (40 Points)

In order to prepare the "The First National ACM School Contest" (in 2020), the major of the city decided to provide all the schools with a reliable source of power (The major is really afraid of blackouts). So, in order to do that, power station "PS" and one school (doesn't matter which one) must be connected; in addition, some schools must be connected as well. You may assume that a school has a reliable source of power if it's connected directly to "PS", or to any other school that has a reliable source of power. You are given the cost of connection between some schools. The major has decided to pick out the cheapest connection plans – the cost of the connection is equal to the sum of the connections between the schools. Your task is to help the major find the cost of the cheapest connection plans.

Input

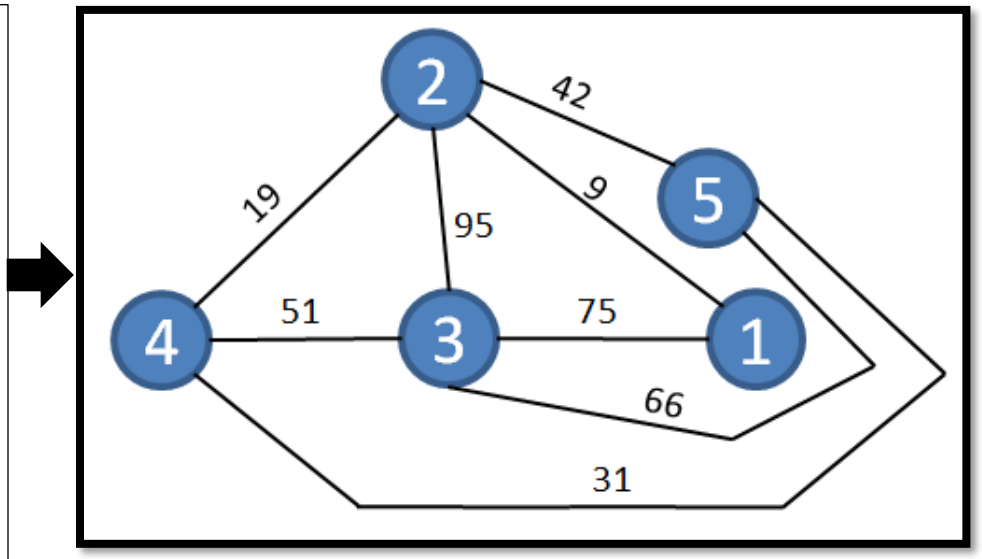
Input file is a txt file that includes the information to build your school connected structure as follows:

- The first line of the input contains two numbers, which are separated by a space, N ($3 < N < 100$) the number of schools in the city, and M the number of possible connections among them.
- Next M lines contain three numbers A_i, B_i, C_i, where C_i is the cost of the connection ($1 < C_i < 300$) between schools A_i and B_i. (The schools are numbered with integers in the range 1 to N).

Here is the sample input txt file examples:

Input 1:

```
5 8

1 3 75

3 4 51

2 4 19

3 2 95

2 5 42

5 4 31

1 2 9

3 5 66
```



Output

Output contains two line. The first line should contain a number as the cost of the cheapest connection plan. And the second line contain the connections inside the cheapest connection plans. You can print edges in any order separated with comma.

Sample Outputs:

Output 1:

```
110

1-2,2-4,4-5,3-4
```

PART II (60 points)

You have just moved into a new apartment and have a long list of items you need to buy. Unfortunately, to buy this many items requires going to many different stores. You would like to minimize the amount of driving necessary to buy all the items you need. Your city is organized as a set of intersections connected by roads. Your house and every store are located at some intersection. Your task is to find the shortest route that begins at your house, visits all the stores that you need to shop at, and returns to your house.

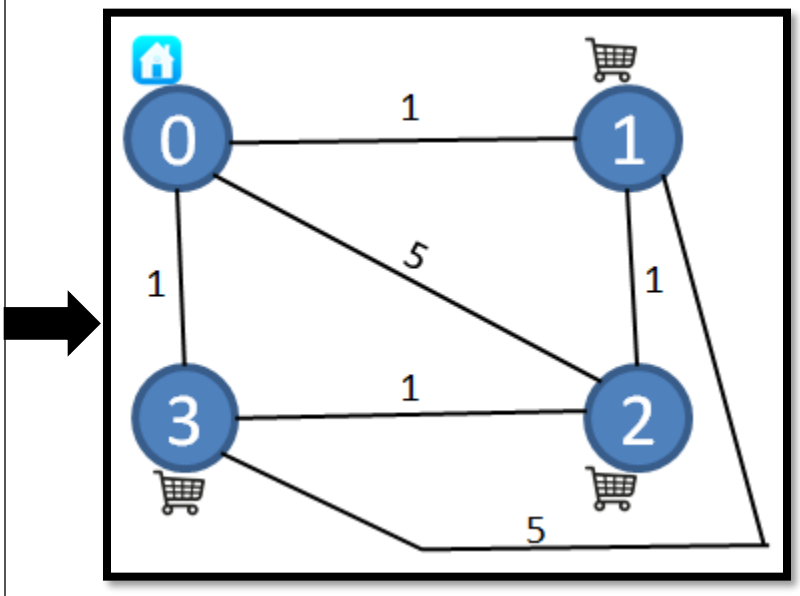
Input

Input file is a txt file that includes the information to build your road structure as follows:

- The first line of input contains the number of N intersections as an integer (The intersections are numbered from 0 to $N - 1$)
- The second line of input contains the number of M roads in the city as an integer. (this integer is between 1 and 100000, inclusive)
- M lines follow, each containing three integers X, Y and D, indicating that the intersections X and Y are connected by a bidirectional road of length D.
- The following line contains a single integer S, the number of stores you need to visit, which is between 1 and ten, inclusive. (The subsequent S lines each contain one integer indicating the intersection at which each store is located and it is possible to reach all of the stores from your house.)

Here is the sample input txt file example:

```
4
6
0 1 1
1 2 1
2 3 1
3 0 1
0 2 5
1 3 5
3
1
2
3
```



Output:

Output a line containing a single integer that is the length of the shortest possible shopping trip from your house, visiting all the stores, and returning to your house.

Hint: Your house is at the intersection numbered 0.

Sample Output:

```
4
```

WHAT TO HAND IN

A zip file containing:

- ➔ The Java source of your programs.
- ➔ The Java sources should be WELL DOCUMENTED as comments, as part of your grade will be based on the level of your comments.

The zip file should be uploaded to Moodle.

IMPORTANT

IMPORTANT NOTES: Do not start your homework before reading these notes!!!

1. You should submit your homework to course Moodle page before the deadline. No hardcopy submission is needed. You should send files and any additional files if you wrote additional classes in your solution as a single archive file (e.g., .zip, .rar).
2. The standard rules about late homework submissions apply (20 points will be deducted for each late day). Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.
3. You ARE NOT ALLOWED to modify the given method names. However, if necessary, you may define additional data members and member functions.
4. For this assignment, you are allowed to use the codes given in our textbook and/or our lecture slides. You ARE NOT ALLOWED to use any codes from somewhere else (e.g., from the internet, other text books, other slides ...).
5. The submissions that do not obey these rules will not be graded.
6. To increase the efficiency of the grading process as well as the readability of your code, you have to follow the following instructions about the format and general layout of your program.
7. Do not forget to write down your id, name, section, assignment number or any other information relevant to your program in the beginning of your Java files. Example:

```
//-------------------------------------------------

// Title: Graph

// Author: Hamid Ahmadlouei

// ID: 2100000000
```

```
// Section: 0

// Assignment: 3

// Description: This class defines a graph

//-------------------------------------------------
```

8. Since your codes will be checked without your observation, you should report everything about your implementation. Well comment your classes, functions, declarations etc. Make sure that you explain each function in the beginning of your function structure. Example:

```
void setVariable(char varName, int varValue)

  //-----------------------------------------------------------------
  // Summary: Assigns a value to the variable whose name is given.
  // Precondition: varName is a char and varValue is an integer
  // Postcondition: The value of the variable is set.
  //-----------------------------------------------------------------
  {
          // body of the function
  }
```

   - Indentation, indentation, indentation...

9. This homework will be graded by your TAs, İbrahim İleri and Hamid Ahmadlouei (ibrahim.ileri@tedu.edu.tr, hamid.ahmadlouei@tedu.edu.tr). Thus, you may ask them your homework related questions through HW forum on Moodle course page. You are also welcome to ask your course instructor Tolga Çapın and Ulaş Güleç for hel

GRADING RUBRICs

PART I (40 points)						
Performance	**Weight**	**Master**	**Advanced**	**Developing**	**Beginner**	**Insufficient**

Element		(4/4)	(3/4)	(2/4)	(1/4)	(0/4)
Information	5%	Information (id, name, section, assignment number) given in each file.	Missing minor details.	Missing few details (e.g. section id).	Only name given.	None.
Documentation	5%	Every class and method has a detailed explanation (pre- and post-conditions, sample I/O, etc).	Most classes and methods have an explanation, but missing in some parts.	Attempted to document the classes and methods, but they are not clear.	Only few comments.	Not even an attempt.
Code Design	5%	Modular source code and code format. Complete submission of classes and methods.	Methods make sense. Includes constructor that initializes carefully.	Uses set/get methods as necessary.	Class does very little; most functions remain in one main (driver) class.	Methods not properly defined.
Abstract Data Type: Graph	10%	Uses graph based data structure for implementation. And provides all functionality.	Uses graph based data structure for implementation. And provides most of expected functionalities.	Implements graph with major deviations from the specification.	Used different data structure from directed graphs to solve this problem.	Not even an attempt.
Functionality	65%	All functions are implemented with no missing functionality. Runs without any crash.	Missing some minor features or minor output problems. Runs without any crash.	Attempted to implement all functions but some of them do not work.	Only few functions are implemented correctly. Compiles but several warnings.	No working solution or does not compile.
Testing: Test data creation & generation	10%	Provided a tester class. Able to identify key test points. Clear on what aspects of the solution are being tested with each set. Able to	Provided a tester class. Able to identify key test points. Clear on what aspects of the solution are being tested with each set.	Provided a tester class. Able to identify key test points.	Sporadic.	No test case presented.

		generate test data automatically when necessary.				

PART II (60 points)						
Performance Element	**Weight**	**Master (4/4)**	**Advanced (3/4)**	**Developing (2/4)**	**Beginner (1/4)**	**Insufficient (0/4)**
Information	5%	Information (id, name, section, assignment number) given in each file.	Missing minor details.	Missing few details (e.g. section id).	Only name given.	None.
Documentation	5%	Every class and method has a detailed explanation (pre- and post-conditions, sample I/O, etc).	Most classes and methods have an explanation, but missing in some parts.	Attempted to document the classes and methods, but they are not clear.	Only few comments.	Not even an attempt.
Code Design	5%	Modular source code and code format. Complete submission of classes and methods.	Methods make sense. Includes constructor that initializes carefully.	Uses set/get methods as necessary.	Class does very little; most functions remain in one main (driver) class.	Methods not properly defined.
Abstract Data Type: Graph	10%	Uses graph based data structure for implementation. And provides all functionality.	Uses graph based data structure for implementation. And provides most of expected functionalities.	Implements graph with major deviations from the specification.	Used different data structure from directed graphs to solve this problem.	Not even an attempt.
Functionality	65%	All functions are implemented with no missing functionality. Runs without any crash.	Missing some minor features or minor output problems. Runs without any crash.	Attempted to implement all functions but some of them do not work.	Only few functions are implemented correctly. Compiles but several warnings.	No working solution or does not compile.

Testing: Test data creation & generation	10%	Provided a tester class. Able to identify key test points. Clear on what aspects of the solution are being tested with each set. Able to generate test data automatically when necessary.	Provided a tester class. Able to identify key test points. Clear on what aspects of the solution are being tested with each set.	Provided a tester class. Able to identify key test points.	Sporadic.	No test case presented.