CMPE 343
Fall 2022
Programming Homework 1

This assignment is due by 23:55 on Friday 2, December 2022.

You are welcome to ask your HW related questions by joining the recitation hours given below:

There will be two Q&A Office Hours on the following days:

- CMPE343-HW3-OfficeHour1: November 23 Wednesday, 18:00-20:00 PM, Zoom ID: https://tedu.zoom.us/j/95503031088
- CMPE343-HW3-OfficeHour2: November 30 Wednesday, 18:00-20:00 PM, Zoom ID: https://tedu.zoom.us/j/97755541656

Note: Please make sure that you have read the HW document well before participating. However, no HW related questions will be accepted except from the above options.

PROGRAMMING TASK

In this part, you must implement your own graph data structure by taking inspiration from your textbook and use it to help to solve problem. You are not allowed to use any external library or .jar file. Any solutions without using graph data structure are not evaluated!

Question 1(25 points):

You are working in a post office, and you need to deliver a letter to an address by using some streets. You have the information of the streets and the distance between one street to another.

In this question, you are expected to deliver the letters to the addresses by using the streets which shorten the distance between the starting point and the delivery address.

In the input, the first line defines the total number of streets, the second line defines the number of connections between the streets, and the rest of the lines define the connections and the distance between the streets. And the last lines define the starting points and the delivery addresses for the letters you need to deliver.

Let assume that you need to deliver the letter from street A to street F by using the streets which shorten the distance.

Here is an example input:

```
6
8
A B 3
A D 7
B C 5
B E 4
C E 2
D E 1
D F 7
E F 10
A B
B C
C D
D E
E F
```

The expected output will be the shortest distance between the starting point to the delivery address and the streets you used.

Here is the output of the given input above:

```
3
6
3
1
8
```

You should use graph data structure to solve this question. **Other solutions that do not use graph will not be evaluated.**

Question 2(25 points):

You use car to go post office by using different roads every day, and you realized some of the roads causes you to be late work because of the traffic lights. You want to eliminate the roads which causes cycles and have long traffic light duration from your route to work.

In the input, the first line defines the total number of roads, the second line defines the number of connections between the roads, and the rest of the lines define the connections and the duration you wait on the traffic lights between the roads.

Let assume that your house is on the road A and post office is on the road F, you need to find the total time you waste on traffic lights when you eliminate some of the roads according to the rules explained above.

Here is an example input:

```
6
8
A B 11
A D 5
B C 12
B D 7
C E 14
C F 22
D E 16
E F 20
A F
```

The first line of the expected output will be the total you waste on traffic lights. The second line defines the number of roads is you will not use, and the rest of the lines defines the roads you will not use to not waste more time on traffic lights.

Here is the output of the given input above:

```
58
3
A B
D E
C F
```

You should use graph data structure to solve this question. **Other solutions that do not use graph will not be evaluated.**

WHAT TO HAND IN

- **You need to upload your code into VPL on LMS for each question.** If you do not upload your code into VPL on LMS, your homework will **not be evaluated.**
- The Java sources should be WELL DOCUMENTED as comments, as part of your grade will be based on the level of your comments.
- You need to upload **maximum-3 pages** PDF report document that explains your own answers for programming task in a clearly readable PA report format (refer to **PA REPORT FORMAT** section).

PA REPORT FORMAT

A programming assignment report is a self-description of a programming assignment and your solution. The report must not be hand-written. You may use a word processor or the on-line editor of your choice and prepare as a PDF document. The report must be grammatically correct and use complete English sentences. Each report should include the following sections, in the order given:

Information (%2.5): This section includes your ID, name, section, assignment number information properly.

Problem Statement and Code Design (%15): Include a brief summary of the problem and/or your sub-tasks to be completed in this assignment. You should show your modular design rationale by creating a structure chart that indicates your top-down, stepwise refinement of the problem solution. You may create the structure chart using available graphical tools like MS PowerPoint, SmartDraw etc.

Implementation and Functionality (%20): Since you have modular source code, you should describe each sub-module (program) in this section. Each sub-module should include names and types of any input/output parameters as well as the pseudocode algorithm that used for completing its task. By this way, you give meaning to each chart boxes from the previous section.

Testing (%7.5): You should provide a tester class that is able to identify key test points of your program. This class should be able to generate additional (apart from the given sample input/output) test data for the purpose of being clear on what aspects of the solution are being tested with each set. This section should also include a description of any program *bugs* that is, tests which has incorrect results. You should write these to describe your tests, summarize your results, and argue that they cover all types of program behavior.

Final Assessments (%5): In this final section, you should briefly answer the following questions:

- What were the trouble points in completing this assignment?
- Which parts were the most challenging for you?
- What did you like about the assignment? What did you learn from it?

GRADING:

- Codes (%50: %25 for Q1 and %25 for Q2)
  - Available test cases evaluation on VPL: %15
  - Hidden test cases evaluation: %15
  - Approach to the problem: %20
- Report (%50: %25 for Q1 and %25 for Q2)
  - Information: %2.5
  - Problem Statement and Code design: %15
  - Implementation, Functionality: %20
  - Testing: %7.5
  - Final Assessments: %5

IMPORTANT

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

1. This assignment is due by 23:55 on Friday, December 2th.

2. You should upload your homework to Moodle before the deadline. No hardcopy submission is needed. You should upload your code files into VPL and your report.

3. 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.

4. You ARE NOT ALLOWED to modify the given method names. However, if necessary, you may define additional data members and member functions.

5. Your classes' name MUST BE as shown in the homework description.

6. The submissions that do not obey these rules will not be graded.

7. 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.

8. 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: Scheduler tester class
// Author: Name/Surname
// ID: 2100000000
// Section: 1
// Assignment: 1
// Description: This class tests the …
//-------------------------------------------------------
```

9. Since your codes will be checked without your observation, you should report everything about your implementation. Add detailed comments to 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
}
```

10. Indentation, indentation, indentation...

11. This homework will be graded by your TAs, Deniz Merve Gündüz, Mehmet Bahadır Aşkın. Thus, you may ask them your homework related questions through office hours. You are also welcome to ask your course instructors İsmail Bora Çelikkale for help.