# CMPE 343
# Spring 2022
# Programming Homework 3

This assignment is due by 23:55 on Friday 22, April 2022.

You are welcome to ask your HW related questions. You should use only one of these options:

1. Moodle Homework **Question Forum**: HW Question-and-Answer (Q&A) Forum on Moodle is always available. Use the "Forum" link at the course Moodle page.
2. Homework **RECITATION HOURS**: There will be two Q&A RECITATION HOURs on the following days:
   - CMPE343-HW3 - OfficeHour1: **April 18, 04:00-05:00 PM,** Zoom ID: https://tedu.zoom.us/j/91886988576?pwd=MXVlQ3B1Y252T3ZpVWYySlZJK1RSZz09
   - CMPE343-HW3 - OfficeHour2: **April 22, 04:00-05:00 PM,** Zoom ID: https://tedu.zoom.us/j/94466817265?pwd=QnJJd0pwRHhIZzk4b3kvZWkzY3RTdz09
   - CMPE343-HW3 – OfficeHour3: **April 25, 04:00-05:00 PM,** Zoom ID: https://tedu.zoom.us/j/92943033994?pwd=dFJFcloxYTB0bVBlNWY4WmxsSXBpUT09

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):**

This is the year that TEDU is hosting Hackathon 2022. To organize the contest a lot of LAN cables will be laid down. **N** students from different cities will be coming to earn the prize of Hackathon winner. You are being asked to come up with a plan such that LAN utilization is minimized, and all computers are connected to each other. LAN will be laid to connect these **N** computers. You will be given information about which computer to be connected to which computer and distance between them. After you have come up with a plan you will be asked **M** queries of the form for which you have to tell the minimum length of LAN used connect computers *u* and *v* according to your plan.

**Input format:**

In the first line there are 3 integers **S, C** and **M** denoting the number of students, the number of connections and the number of queries respectively. Next **C** line contains 3 integers each denoting computer *u* and computer *v* and *d* denoting distance between them. Next **M** line contains 2 integers each denoting computer *u* and computer *v.*

Assume that there will be 6 students for the contest and there will be 9 LAN connections between the computers of the students and 5 queries will be asked for the distance of the connections between the computer *u* and computer *v* after you apply your connection plan.

Here is an example input:

```
6 9 5
1 2 5
1 3 4
1 4 6
2 3 2
3 4 1
2 5 9
3 5 3
4 6 7
5 6 10
1 2
2 3
3 4
4 5
5 6
```

**Output format:**

For each query **M,** you need to give the minimum length of cable used to connect computer *u* and computer *v.* According to example input given above, you need give the length of the cable between the computer 1 and 2, 2 and 3 and so on till 5 and 6.

Here is an example output:

```
6
2
1
4
11
```

**Question 2(25 points):**

After Hackathon 2022 contest, TEDU will host a conference for the participants and the auditoriums in the conference area in the form of a graph having **N** nodes and **M** edges. The graph does not have self-loops or multiple edges. Each node represents an auditorium and has a capacity of seating available. We can reach from node *u* to node *v* incurring a cost of *w* units. Conference area has a walking distance from the entrance of the campus **W** which is same for all participants.

There are **K** identical participants entering the conference area, each participant enumerated with a distinct number from **1** to **K**. The participants enter in their natural order, that is, participant number 1 enters, then participant number 2, then 3 and so on till participant number **K** according to the capacity of the auditoriums.

For each participant, you have to print the minimum total distance that the participant must get over to the auditoriums. Here, total distance includes **cost of the path taken to reach the conference area from the entrance of the campus W and the distance from the conference area to an auditorium which has available seating**. It is guaranteed that you can reach any slot from any other slot. **All participants entering the auditoriums coming from the entrance of the campus and you need to direct them starting from the 1st auditorium.**

**Input Format:**

The first line consists of three integers **N, M** and **F** denoting number of nodes (which represents auditoriums), number of edges and distance of auditorium from the conference area respectively. The second line consists of **N** space separated integers denoting the seating capacity of each auditorium. Following **M** lines contain three space separated integers each: u, *v* and *w*, denoting we can reach from node *u* to node *v* incurring a cost of *w* units. The last line of input contains an integer **K** denoting the number of participants enter the conferences.

Assume that there are 5 auditoriums, 4 connections between them and 10 distance from the entrance of the campus to the conference area. The capacity of each auditorium is 1, 2, 1, 1 and 2. In the next 4 lines, the distance between these auditoriums is given and the number of students will be joining the auditoriums is 5. You need to give the total distance they need to walk for the auditorium.

Here is an example input:

```
5 4 10
1 2 1 1 2
1 2 2
4 5 1
3 4 2
1 3 1
5
```

**Output is given on the next page!**

**Output Format:**

Print **K** space separated integers denoting answer for each participant. ith integer in the space separated integers denotes answer for ith participant number. If it is not possible to enter an auditorium print −1 for that participant.

Here is the output of the given input above:

```
10 11 12 12 13
```

## WHAT TO HAND IN

A zip file for both parts containing:

- The Java sources for your program.
- The Java sources should be WELL DOCUMENTED as comments, as part of your grade will be based on the level of your comments.
- You can test your Java source files on available Moodle VPL environment to ensure your solution's correctness before submitting. VPL simply tests your program's output by checking against given sample input.
- A **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, 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. Also, you should add your performance comparison, part II, here.

**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, April 29th**.

2. You should upload your homework to Moodle before the deadline. No hardcopy submission is needed. You should upload files and any additional files if you wrote additional classes in your solution as a single archive file (e.g., zip, rar).

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, Bedrettin Çetinkaya, Deniz Merve Gündüz. Thus, you may ask them your homework related questions through HW forum on Moodle course page. You are also welcome to ask your course instructors Tolga Çapın for help.