CMPE 343
Spring 2023
Programming Homework 1

This assignment is due by 23:59 on Sunday 9, April 2023.

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-HW1-OfficeHour1: March 29, 16:00-18:00, Zoom ID: https://tedu.zoom.us/j/93270367009
- CMPE224-HW1-OfficeHour2: April 5, 16:00-18:00, Zoom ID: https://tedu.zoom.us/j/95367510902

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 a flight operations manager in a flight company and there are *N* cities and *M* undirected flight routes you need to organize in the country. Each city has an airport and each airport can work as layover. The airport will be in two states, loading and running. In loading state, luggage is loaded into the planes. In the running state, planes will leave the airport for the next city. All the airports will switch their states from loading to running and vice versa after every *T* minutes. At an airport, if its state is loading, you have to wait for it to switch its state to running. At the beginning, all the airports are in running state. The time taken to travel through any flight route is *C* minutes. Find the lexicographically smallest path which will take the minimum amount of time (in minutes) required to move from city *X* to city *Y*.

In the input, the first line contains *4* space separated integers, *N*, *M*, *T* and *C*. *N* denotes the number of cities we have, *M* denotes the number of connections between the *N* cities, *T* denotes the time required by airports to change their states and *C* denotes that the time for travelling one city to another. Next *M* lines contains two space separated integers each, *U* and *V* denoting that there is a bidirectional road between city *U* and city *V*. Next line contains two space separated integers, *X* the city we start to travel and *Y* the city we want to reach at the end.

In the first line it is given that we have 5 cities and 5 bidirectional roads between them. For this case, time required the airports to change their state is 3 minutes and travel through any flight route is 5 minutes. The next lines gives the connections between the cities and the last line gives which city is the starting point and which city is the end point for this case.

Sample Input:

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

In the output, the first line print an integer *K*, denoting the number of city you need to go through to reach city *Y* from the city *X*. In next line, print *K* space separated integers denoting the path which will take the minimum amount of time (in minutes) required by to move from city *X* to city *Y*. In the last line, print the total time through the path from city X to city Y. There can be multiple paths. Print the lexicographically smallest one and then the total time for each path at the end.

The output for the above inputs as follows. Please check your program with this input as well as the others that you will create. Please note that we may use other inputs when grading your assignments.

Sample Output:

```
3
1 2 5
11
```

Question 2(25 points):

You are a tour guide in a coastal city and you need to organize a ship tour for your guests. There are *N* islands around the city and *M* undirected paths between the islands. The ship tour you will organize should start from island *X* and include island *Y* and return the beginning point at the end of the tour.

In the input, the first line contains *2* space separated integers, *N*, *M*. *N* denotes the number of islands we have, *M* denotes the number of connections between the *N* islands. Next *M* lines contains two space separated integers each, *U* and *V* denoting that there is a bidirectional road between island *U* and island *V*. Next line contains two space separated integers, *X* the island we start tour and *Y* the island tour should include.

In the first line it is given that we have 5 islands and 6 bidirectional paths between them. The next lines gives the connections between the islands and the last line gives which island is the starting point and which island should be included to the tour for this case.

Sample Input:

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

In the output, print *K* space separated integers denoting the path which the tour starts from island *X* and includes island *Y*. There can be multiple paths. Print the lexicographically smallest one and then the total time for each path at the end.

The output for the above inputs as follows. Please check your program with this input as well as the others that you will create. Please note that we may use other inputs when grading your assignments.

Sample Output:

```
1 2 3 4
```

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 handwritten. 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 23:59 on Sunday, April 9th.

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, Elif Ünal and Enes Arslan. 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 Kurtuluş Çapın and Ulaş Güleç for help.