CMPE 343
Spring 2023
Programming Homework 2

This assignment is due by 23:55 on Tuesday 2, May 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-HW2-OfficeHour1: April 20, 06:00-07:00 PM, Zoom ID:

  https://tedu.zoom.us/j/94973263513?pwd=VUJodm9xZU0vRUQ0RFZXWEgzS2VhZz09

- CMPE224-HW2-OfficeHour2: April 27, 06:00-07:00 PM, Zoom ID:

  https://tedu.zoom.us/j/99377804472?pwd=ZWtnTUM4YnFGM1FFYkptNzdLVmI1UT09

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

PROGRAMMING TASK

In this part, you must implement your own graph data structure by taking inspiration from your textbook and using it to help to solve the 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 started as a software developer in the Uber company. In this company, a special ride network is created every day. And your first task is to prove whether you can keep this company's ride network in a tree structure. But the developer who left the company before you has already designed this network using a directed graph. Your task is to measure whether this directed graph can be a tree. After some research, you found that for a directed graph to be a tree, it must have

one less number of edges than the number of vertices, it must be connected, and there must be no cycles in the graph. Now prove to your team leader that you can solve this problem.

Note: Here you must use the symbol chart data structure when retrieving taxi pickups from the user.

Sample Input:

```
Enter the number of taxi pickups:
4
Enter the number of taxi rides:
5
Enter the taxi rides:
Cayyolu Emek
Emek Kızılay
Emek Yenimahalle
Kızılay Cayyolu
Yenimahalle Kızılay
```

The person who will use your program must first enter the number of taxi pickups in the ride network. Then it should enter how many taxi rides there are connecting these taxi pickups. Finally, it should specify line by line from which taxi pickup to which one can go. For example, in the first line, it is stated that there is a one-way departure from Çayyolu to Emek.

Sample Output:

```
Cayyolu:  Emek
Emek:  Yenimahalle Kızılay
Kızılay:  Cayyolu
Yenimahalle:  Kızılay
This ride network cannot be kept in a tree structure.
```

In the program output, show the user to which taxi pickups a taxi in this ride network can go from its location. For example, in line 1, it is stated that there are only departures from Çayyolu to Emek. And in the last line, print the information on whether this ride mesh can be kept in a tree structure.

Question 2 (25 points):

All farmland in a town will be harvested by a company. The company first needs to calculate the minimum cost of navigating between the farms. Since these farmlands are all next to each other, information is collected from the landowners in the town and kept in a grid structure with N rows and M columns. You are asked to write a program for this company. This program uses the **MST (Minimum Spanning Tree)** algorithm to output a single integer expressing the minimum cost of good trip in a grid structure.

Let your program first get the number of test cases from the user. For each test case, ask for the number of rows and columns in the grid. Then calculate the cost of an edge e connecting any two cells with values a and b as follows:

$E(e)=a \oplus b$ *(where $\oplus$ denotes the bitwise xor operation of integers a and b)*

We define good trip between two cells (u1,v1) and (u2,v2) as a trip starting at cell (u1,v1) and ending at cell (u2,v2) while visiting every cell of the grid at least once. For a given edge e, if you visit this edge T(e) times then the cost of the trip is:

$\sum_{(\forall e)} (E(e) * (T(e) / 2)$ *(here, T(e) / 2 is the ceiling of the result of the division of T(e) by 2)*

Now please find the minimum path cost of harvesting all farmland for the given starting cell (u1,v1) and ending cell (u2,v2).

Sample Input:

```
1
2 2
1 1 2 1
1 2
2 2
```

Here, the first line is the number of test cases and the next line is the grid size for that test case (N X M).

The line expressed by line 3 for this sample case is will consist of four integers, r1,c1,r2,c2, denoting the coordinates (r1,c1) of the starting cell and (r2,c2) of the ending cell.

Finally, N lines will follow, each containing M integers, denoting the values in the grid, such that the j.value in the i.row will denote the number in the cell (i, j) of the grid.

Sample Output:

```
3
```

It outputs a single integer indicating the minimum cost of trip for each of the test cases, one after the other.

Sample_2 Input:

```
2
3 1
3 1 3 1
4995
1461
4998
1 3
1 2 1 3
1 3 4
```

Sample_2 Output:

```
11369
9
```

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 23:55 on Tuesday, May 2nd.

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 the HW forum on Moodle course page. You are also welcome to ask your course instructors Tolga Kurtuluş Çapın and Ulaş Güleç for help.