Project 10: You are to implement the Graph coloring problem using the two greedy methods taught online. The algorithm steps for the two methods are also posted on BB.

This will be your last project for the semester!

Your will be given three test data: data1, data2, and data3

- Run your program using method-1 on data1

- Run your program using method-2 on data1

- Run your program using method-1 on data2

- Run your program using method-2 on data2

- Run your program using method-1 on data3

- Run your program using method-2 on data3

Strictly, run your program in the above sequencing.

Include in your pdf hard copies:

- cover page

- source code

- method-1 on data1

- method-2 on data1

- method-1 on data2

- method-2 on data2

- method-1 on data3

- method-2 on data3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Language: C++

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Project points: 12 pts

Due Date: Soft copy and pdf hard copies: 5/8/2020 Friday before midnight

+1 early submission: 5/5/2020 Tuesday before midnight

- 1 pt 1 day late: 5/6/2020 Saturday before midnight

- 2 pts 2 days late: 5/7/2020 Sunday before midnight

-12 pts: after 5/7/2020 Sunday after midnight

\*\*\* Name your pdf file using the same naming convention as given prior

\*\*\* All on-line submission MUST include Soft copy and pdf hard copy in the same email with correct file names; otherwise, it would not count as submission.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I. Inputs:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1) inFile1 (use argv[1]): a text file contains a list of bi-directional edges,

representing an undirected graph, G=<N, E>. A bi-directional edge in the data file should be treated as having two edges. For example, an undirected edge <3, 5> in the data file, means one directed edge from 3 to 5 and another directed edge from 5 to 3.

The input file format is as follows:

The first number in inFile1 is the number of nodes in the graph;

then follows by a list of undirected edges <ni, nj>.

For example

8 // 8 nodes in the graph

1 2

4 1

2 4

5 2

2 6

3 4

:

:

:

2) whichMethod (use argv[2]): // Let the program knows which method to use: 1 means use method-1; 2 means use method-2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

II. Outputs:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1) outFile1 (use argv[3]) : for the output of the color assignments of nodes in the graph

use the following format:

Method 1 (or 2) was used to color the graph. // The first output text-line

Below is the result of the color assignments. // The second output text-line

// Provide a blank text-line

8 // 8 nodes in the graph

1 1 // node 1 colored with 1st color

2 3 // node 2 colored with 3rd color

:

:

:

2) outFile2 (use argv[4]): output the content of your data structure of the graph, nicely formatted

(Do the best you can, if TA think you are doing an outstand job, you may get extra point,)

3) outFile3 (use argv[5)]: For all your debugging prints, if you need.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

III. Data structure:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

You are to design your own data structures (class and methods).

Remark: a graph can be represented by an adjacent matric or a hash table (an array of linked list)

\*\*\* You may NOT use any C++ build-in data structure, but to write your own data structures.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IV. main (…)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Insert all needed output prints to outFile1, outFile2, and outFile3, where they deem fit.

Step 0: inFile 🡨 open argv[1]

outFile1, outFile2, outFile3 🡨 open from argv

Step 1: whichMethod 🡨 from argv[2]

step 2: case of whichMethod

case 1: Method1 (inFile, …)

case 2: Method2 (inFile, …)

default: error message

Step 3: close all files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

V. Method1 (…)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copy from the lecture notes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

VI. Method2 (…)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copy from the lecture notes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

All other methods you have designed