CS 3305/W02: Data Structures

Assignment 11 - Graphs   Due 11/17/2022 @ 11:59 PM   (100 points total)

GENERAL SUBMISSION REQUIREMENTS

Upload all files individually as specified, not as zip files, to Assignments in D2L. Do not email files.
Make sure your program compiles, runs and produces the correct output.
Ensure you have the correct file name(s), and author header, as specified in the Assignment.
Always use meaningful labels for prompts, inputs, and outputs.
Always use comments, indentation and whitespace as shown in examples.
Note: Never hard-code test data in the test program, unless explicitly stated in the assignment.
Always allow the user to enter the test data using a menu option.

Assignment 11 – PART 1   DFS   (100 points) :

There is only one Part to this assignment, Programming Exercise 28.3 from the Liang textbook, end of the Chapter Programming Exercises, re-printed below.

Implement Depth First Search (DFS) using a stack. The depth-first search algorithm described in Listing 28.8 uses recursion. Design a new algorithm without using recursion. First, describe it using pseudocode and copy that pseudocode into the assignment submittal. Next, implement it by defining a new class named UnweightedGraphWithNonrecursiveDFS that extends UnweightedGraph and overrides the [Depth First Search] dfs method.

ISSUE – give them graph data and tell them they can hard code the data

ISSUE – to override a method they have to sift thru book code and figure out how to do it – they were not overriding method correctly

ISSUE there are dependences – the DFAS method returns a tree

ISSUE - UnweightedGraph extends AbstractGraph (which is where DFS method is found).
DFS returns a tree in the nestedclass.java – because you are supposed to return a tree you must look thru the constructor and figure out what parameters are needed and how to create the method. To properly override the method, it is not enough to print the search order.

NOTE: when you review Listing 28.8 you will notice that there are two dfs methods. You need to override the method dfs(int v), not the other one with the longer list of parameters.

Do not forget to include author header in each submitted file as shown, no header, no points!

```
// Name:         <your name>
// Class:        CS 3305/ put your section number after the /
// Term:         Fall 2022
// Instructor:   Sharon Perry
// Assignment:   11-Part-1-DFS
```

Capture a **READABLE** screenshot(s) of your program output and paste into a word/pdf document. Readable means readable!   Screenshots ***should not be an entire desktop*** – use some type of snipping tool. After your output screenshots, copy and paste the pseudo code, and then the source code for your program into the word/pdf doc.   Save doc as a file named

LastName-A11-Part-1-DFS.docx or .pdf.   Last step is to upload word/pdf and .java files to D2L.

SUBMIT YOUR OWN CODE – Code copied from the internet will receive a score of zero.

MAKE SURE YOUR CODE HAS COMMENTS ! We are getting submissions without comments in the code. No comments = (-20) points *per Part of the assignment*