CS 324 Homework Assignment 5

Due: 11:59pm, Thursday, November 23rd

This assignment is scored out of 70. It consists of 5 questions. When you submit, you are required to create a folder with your name (Last name first, then First name), CS324, HW5, e.g., LastName_FirstName_CS324_HW5. Type your answers into a text file (**only `.txt`, `.doc`, and `.pdf` file formats are accepted**) and save it in this folder. Put all your Java programs (**`*.java`**) as well as output files in the same folder. Zip this folder, and submit it as one file to Desire2Learn. Do not hand in any printouts. Triple check your assignment before you submit. **If you submit multiple times, only your latest version will be graded and its timestamp will be used to determine whether a late penalty should be applied.**

Short Answers

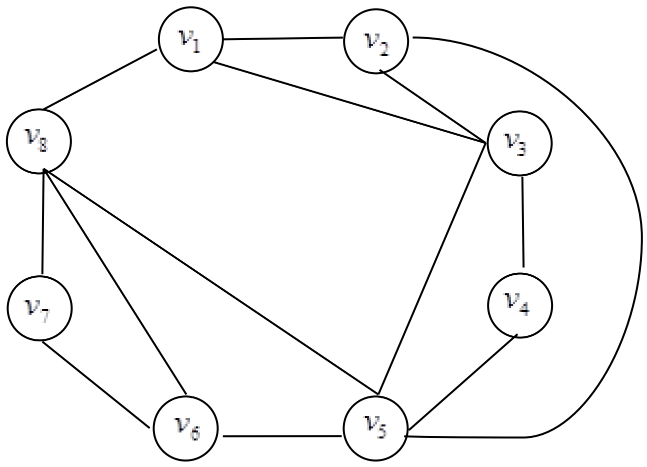
P1. (10pts) Use the backtracking technique in Example 5.1 on page 208-209 in the textbook to find the **first** solution to the 5-Queens problem in the Excel document "**`nQueens.xlsx`**", sheet **P1**. The first two queens are already placed so you can follow the rules to fill the rest of the chessboards. Your first solution should appear on Chessboard 14 or 15.

P2. (10pts) Use the *n*-Queens backtracking algorithm (Algorithm 5.1 on page 212-213 in the textbook) to find the **first two** solutions that are printed out given that the queens in the first three rows cannot be moved (you will start the algorithm in row 4). The chessboard is in the Excel document "**`nQueens.xlsx`**", sheet **P2**. You are NOT required to show the tracing steps.

When you submit your assignment, do not forget to include this Excel file!

P3. (10pts) Can Dijkstra's algorithm be used to find the shortest paths in a graph with some negative weights? **Draw a graph to support your answer.**

P4. (10pts) Trace through the 3-coloring backtracking algorithm on the following graph and give the **first** solution. Using **R**, **G**, **B** to denote color 1, color 2, and color 3, respectively, list the color of the vertices, beginning at v_1. For example, v_1: R, v_2: G, v_3: B ....., v_8:x (**The first three vertices must be colored in this order**).



<u>Programming Questions</u>

P5. (30pts)

In this programming question, you are required to implement the *n*-Queens backtracking algorithm (Algorithm 5.1 on page 212-213 in the textbook). **Instead finding all solutions, you are required to find only the first solution and record the number of time that the `queens` method is called to find this solution.**

<u>a. Completing the `Homework5` class</u>

You are provided with two files "**`Homework5.java`**" and "**`TestHomework5.java`**". You are required complete the **`queens`** method and the **`promising`** method in the former file to implement the modified *n*-Queens algorithm:

`void queens(int i)` – This is a recursive method that takes an **`int`** parameter (the starting queen index) and **finds the first solution** to the *n*-Queens problem. When the solution is found, instead of using a **`for`** loop to print out the content of the **`col`** array, you are required to print out the complete chessboard (see sample output for format) as well as call the **`verifySolution`** method and pass **`col`** into it to verify if the current solution is correct. Make sure you only print out the first solution and the **`numCalls`** instance variable contains the correct number of times that the **`queens`** method is called to find this solution. You can use the instance **`boolean`** variable **`found`** in your implementation if necessary.

`boolean promising(int i)` – This method takes an **`int`** parameter which is the index of a queen and checks if placing it at **`col[i]`** can lead to a solution.

Note that you are only supposed to touch the above methods. You are NOT allowed to create any other methods, instance variables, or make any changes to methods other than the above method or files other than "`Homework5.java`". Points will be taken off if you fail to follow this rule.

<u>b. Code Testing</u>

You are provided with a test driver implemented by "**`TestHomework5.java`**" (**Do not make any changes to this file!**) so there is no need to write your own.

Once you have completed the above method, you can run the test. You should create a plain text file named "**`output.txt`**", copy and paste the output (if your code crashes or does not compile, copy and paste the error messages) to this file and save it.

<u>Grading Rubrics:</u>

Code does not compile: -10
Code compiles but crashes when executed: -5
Changes were made to things other than the required methods: -5
Has output file: 5
`queens` was correctly implemented: 15
`promising` was correctly implemented: 10

Sample Output:

```
Test 1 - queens(3):
 Expected: # calls = 19
 Yours: # calls = 19

Test 2 - queens(4):
- Q - -
- - - Q
Q - - -
- - Q -
This is a valid solution.

 Expected: # calls = 27
 Yours: # calls = 27

Test 3 - queens(6):
- Q - - - -
- - - Q - -
- - - - - Q
Q - - - - -
- - Q - - -
- - - - Q -
This is a valid solution.

 Expected: # calls = 172
 Yours: # calls = 172
```