CE Undirected Graphs CODE

MISSING 10 Possible Points

| 11/14/2022

Attempt 1

IN PROGRESS
Next Up: Submit Assignment

Add Comment

Unlimited Attempts Allowed

Details

ComGraphs

CE: Undirected Graphs CODE

Learning Objectives

- Familiarize yourself with the class DepthFirstPaths
- Deepen your understanding of the recursive depth-first algorithm
- Write code to allow a comparison between your results of tracking the code with the actual values.

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753065)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753071)

Overview

This CE consists of two parts.

In Part 1, you will deepen your understanding of the recursive depth-first algorithm dfs by stepping through it and tracking the changing values of the internal arrays marked and edgeTo.

In Part 2, you will write code that allows you to compare the values you came up with in part 1 with the actual values.

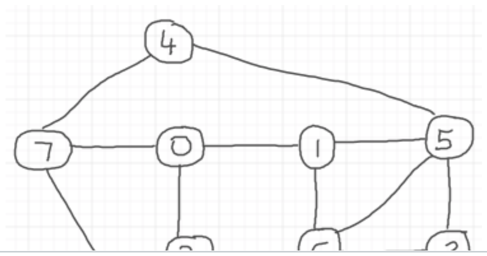
Instruction

Part 1

Use the recursive depth-first algorithm dfs and the graph shown below to identify the shortest paths from the **source vertex 2** to all reachable vertices.

Draw a table similar to the one below including the initial values of the arrays marked and edgeTo.
Update the values of the arrays as you step through the algorithm and visit all the vertices starting with vertex 2.

```
tiny undirected graph:
```



```
Adjacency List:
0: 7 -> 2 -> 1
1: 6 -> 5 -> 0
2: 9 -> 0
3: 8 -> 6 -> 5
4: 7 -> 5
5: 6 -> 4 -> 3 -> 1
```

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753065)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753071)

```
// depth first search from v
private void dfs(Graph G, int v){
   marked[v] = true;
   for (int w : G.adj(v)) {
      if (!marked[w]) {
         edgeTo[w] = v;
         dfs(G, w);
      }
   }
}
```

v	marked[]	edgeTo[]
0	false	0
1	false	0
2	false	0
3	false	0
4	false	0
5	false	0
6	false	0
7	false	0
8	false	0
9	false	0

Identifying the path
When you are done filling in the table, the column below edgeTo shows how the vertex v (which corresponds to the index of the array) can be reached.
E.g. If you wonder, how vertex 8 can be reached starting at 2 (source vertex), look at the value of edgeTo[8]. In our case, this should be 9. That means we can reach 8 coming from 9. Now we have to find out, how can we get to 9. We look up edgeTo[9] and that happens to be 2. This means, in order to get from 2 to 8 we can go from 2 to 9 to 8.

Which path did the dfs algorithm find from 2 (source) to 5?
Write it down. We'll need it later again.

Part 2

This second part of the CE uses code to identify the paths from the source vertex to all other vertices that can be reached. It allows you to compare and validate the correctness of the results you came up with in Part 1 of the exercise.

- Create a package called **graphUndirected** that includes a class **DepthFirstPathsModified**.

(https://slcc.instructure.com/courses/817632/files/135713844/download?download_frd=1)

- In the main method, use local variables to specify the graph file and the source vertex rather than command-line arguments. The graph file should be TinyUndirectedGraph.txt and the source vertex should be vertex **2**.
- Run the program. At this point, the output should look like the output listed under 'Original Output.'
- Modify the main method to accomplish the following:
- Start by printing the adjacency list (see output)
- Print the content of the arrays marked and edgeTo side by side in straight columns. Start each row with the index (see output)
  FYI:
  The API of the class is not designed to publicly reveal this information.
  Because main is declared inside the class itself (in our case DepthFirstPathModified) we do have access to the fields - even though they are declared private.

Self-assessment of Part 1:

1. Compare the table you filled in in part 1 with the values of the arrays marked and edgeTo. It should be the same.
2. Compare the path you wrote down in part 1 with the path from 2 to 5 listed under "Original Output."
3. If the table or the path from part 1 don't match the expected result, revisit part1.

Expected Output

```
Adjacency List:
---------------
0: 7 -> 2 -> 1
1: 6 -> 5 -> 0
2: 9 -> 0
3: 8 -> 6 -> 5
4: 7 -> 5
5: 6 -> 4 -> 3 -> 1
```

< Previous (https://slcc.instructure.com/courses/817632/modules/items/18753065)

Submit Assignment

Next > (https://slcc.instructure.com/courses/817632/modules/items/18753071)

```
   ------ ------
0  true    7
1  true    0
2  true    0
3  true    8
4  true    5
5  true    6
6  true    3
7  true    4
8  true    9
9  true    2

Original Output:
---------------
2 to 0 : 2-9-8-3-6-5-4-7-0
2 to 1 : 2-9-8-3-6-5-4-7-0-1
2 to 2 : 2
2 to 3 : 2-9-8-3
2 to 4 : 2-9-8-3-6-5-4
2 to 5 : 2-9-8-3-6-5
2 to 6 : 2-9-8-3-6
2 to 7 : 2-9-8-3-6-5-4-7
2 to 8 : 2-9-8
2 to 9 : 2-9
```

Submission

Create a screen recording following the guidelines for lab recordings (https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings) .
Part 1 is self-assessed. It should not be included in the video.
The video should be **30 - 60 seconds** long.
Post the video.

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753065)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753071)

(https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753055)

(https://slcc.instructure.com/courses/817632/modules/items/18753057)

(https://slcc.instructure.com/courses/817632/modules/items/18753058)

(https://slcc.instructure.com/courses/817632/modules/items/18753059)

(https://slcc.instructure.com/courses/817632/modules/items/18753061)

(https://slcc.instructure.com/courses/817632/modules/items/18753062)

(https://slcc.instructure.com/courses/817632/modules/items/18753063)

(https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753065)

(https://slcc.instructure.com/courses/817632/modules/items/18753067)

Choose a submission type

Text

Upload

Office 365

More

Previous (https://slcc.instructure.com/courses/817632/modules/items/18753065)

Submit Assignment

Next (https://slcc.instructure.com/courses/817632/modules/items/18753071)