CSCI 3104 Spring 2022
Instructor: Profs. Chen and Layer

Quiz 5 - BFS/DFS

Due Date ..... February 11
Name ..... **Julia Troni**
Student ID **109280095**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

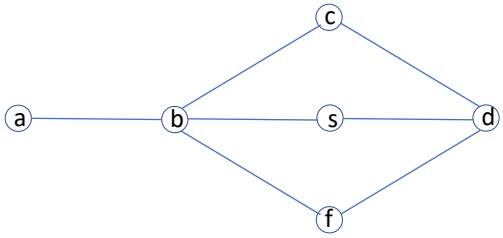
- My submission is in my own words and reflects my understanding of the material.
- Any collaborations and external sources have been clearly cited in this document.
- I have not posted to external services including, but not limited to Chegg, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

I agree to the above, Julia Troni. □

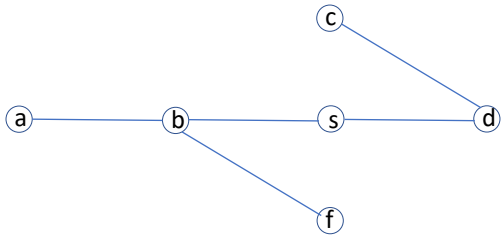
3 Standard 5- BFS/DFS

3.1 Problem 2

Problem 2. Consider the following graph with the source node s:



(i). Is it possible to obtain the following tree using BFS? Clearly justify your answer.



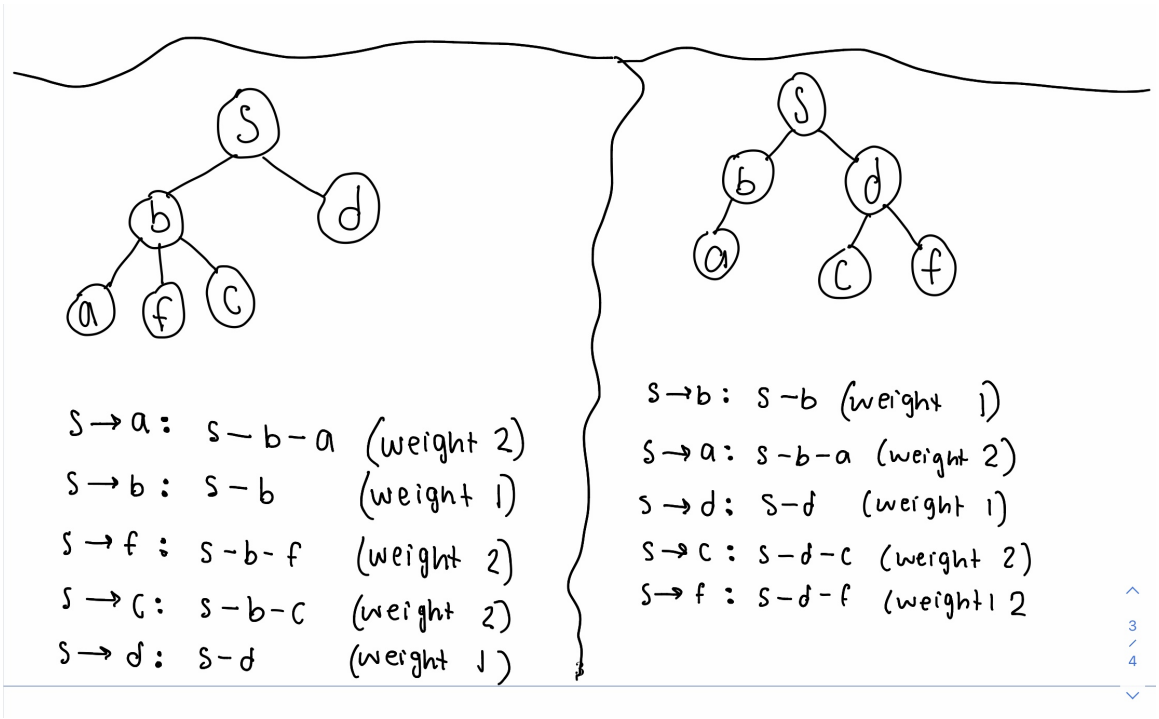
(ii). Is it possible to obtain a shortest path tree using BFS? Clearly justify your answer; and if your answer is yes, give such a shortest path tree obtained using BFS. Here the length of a path is defined as the number of edges on the path.

Answer. **Part** i**:** No, it is NOT possible to obtain that tree using BFS. BFS examines all unvisited neighbors of the current vertex before exploring vertices further away. In this way, BFS would produce a graph where c, a, f, s are the neighbors of b and s is the only neighbor of d. OR c, f, s are the neighbors of d, and a, s are the neighbors of b. Hence, it is impossible to produce that graph with BFS. This is due to the way in which BFS explores vertices, which is described below

A basic BFS algorithm uses the following technique: begin by initializing a state array to track if a vertex has been visited and a predecessor array that will track the parent node along the path. Set all vertices to NEW (unvisited) and all predecessors to NULL. Make a FIFO queue with the start vertex and that predecessor, which is NULL. Then while the queue is not empty, dequeue a vertex, if it is marked as NEW, set it to OLD and save the predecessor. Then for each neighbor of that vertex, add it to the queue.

Part ii**:** Yes BFS does obtain the shortest path tree because this is an unweighted, connected graph. BFS uses the algorithm described above. So, in this way BFS traverses all the connected elements of graph, and due to the FIFO queue, it is traversing layer by layer and hence finding the shortest path.

The following image shows two possible outputs of BFS, both are shortest path trees. Note that while the paths may be different, the total weight to each vertex is the same, and it is the shortest possible.


S
b
d
a
f
c
S → a : s − b − a (weight 2)
S → b : s − b (weight 1)
S → f : s − b − f (weight 2)
S → c : s − b − c (weight 2)
S → d : s − d (weight 1)
S
b
d
a
c
f
S → b : s − b (weight 1)
S → a : s − b − a (weight 2)
S → d : s − d (weight 1)
S → c : s − d − c (weight 2)
S → f : s − d − f (weight 2