Assignment No-2 Title:Breadth-First Search(BFS)

CSE-0408 Summer 2021

Name:Sabrina Oyshe
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
email address:sabrinaoyshe188@gmail.com

***Abstract*—Here i was working Breadth-First Search(BFS) and I solved Problem using C++ language.**

***Index Terms*—Here I mostly used in My report C++ language and Code-block code editor.**

I. Introduction

BFS is a traversing algorithm where you should start traversing from a selected node (source or starting node) and traverse the graph layerwise thus exploring the neighbour nodes (nodes which are directly connected to source node). You must then move towards the next-level neighbour nodes.

II. Literature Review

BFS and its application in finding connected components of graphs were invented in 1945 by Konrad Zuse, in his (rejected) Ph.D. thesis on the Plankalkül programming language, but this was not published until 1972. It was reinvented in 1959 by Edward F. Moore, who used it to find the shortest path out of a maze,[5][6] and later developed by C. Y. Lee into a wire routing algorithm (published 1961). In 2012 Farhad S. et. al. [4] proposed new resolution for solving N-queens by using combination of DFS (Depth First Search) and BFS (Breadth First Search) techniques.

III. Proposed Methodology

Here i Discuss BFS Algorithm:

A standard BFS implementation puts each vertex of the graph into one of two categories:

1. Visited

2. Not Visited

The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

The algorithm works as follows:

1.Start by putting any one of the graph's vertices at the back of a queue.

2. Take the front item of the queue and add it to the visited list.

3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the back of the queue.

4.Keep repeating steps 2 and 3 until the queue is empty.

The graph might have two different disconnected parts so to make sure that we cover every vertex, we can also run the BFS algorithm on every node

IV. BFS Algorithm Applications

1.To build index by search index

2. For GPS navigation

3.Path finding algorithms

4.In Ford-Fulkerson algorithm to find maximum flow in a network

5.Cycle detection in an undirected graph In minimum spanning tree

V. Solving Method

Instill Graph

VI. Solved Graph

Acknowledgment

I would like to thank my honourable**Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.

References

[1] Borges, Paulo HR, et al. "Carbonation of CH and C–S–H in composite cement pastes containing high amounts of BFS." Cement and concrete research 40.2 (2010): 284-292.

[2] Borges, P. H., Costa, J. O., Milestone, N. B., Lynsdale, C. J., Streatfield, R. E. (2010). Carbonation of CH and C–S–H in composite cement pastes containing high amounts of BFS. Cement and concrete research, 40(2), 284-292.

[3] Borges, Paulo HR, Juliana O. Costa, Neil B. Milestone, Cyril J. Lynsdale, and Roger E. Streatfield. "Carbonation of CH and C–S–H in composite cement pastes containing high amounts of BFS." Cement and concrete research 40, no. 2 (2010): 284-292.

[4] Borges, P.H., Costa, J.O., Milestone, N.B., Lynsdale, C.J. and Streatfield, R.E., 2010. Carbonation of CH and C–S–H in composite cement pastes containing high amounts of BFS. Cement and concrete research, 40(2), pp.284-292.

[5] Borges PH, Costa JO, Milestone NB, Lynsdale CJ, Streatfield RE. Carbonation of CH and C–S–H in composite cement pastes containing high amounts of BFS. Cement and concrete research. 2010 Feb 1;40(2):284-92.

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

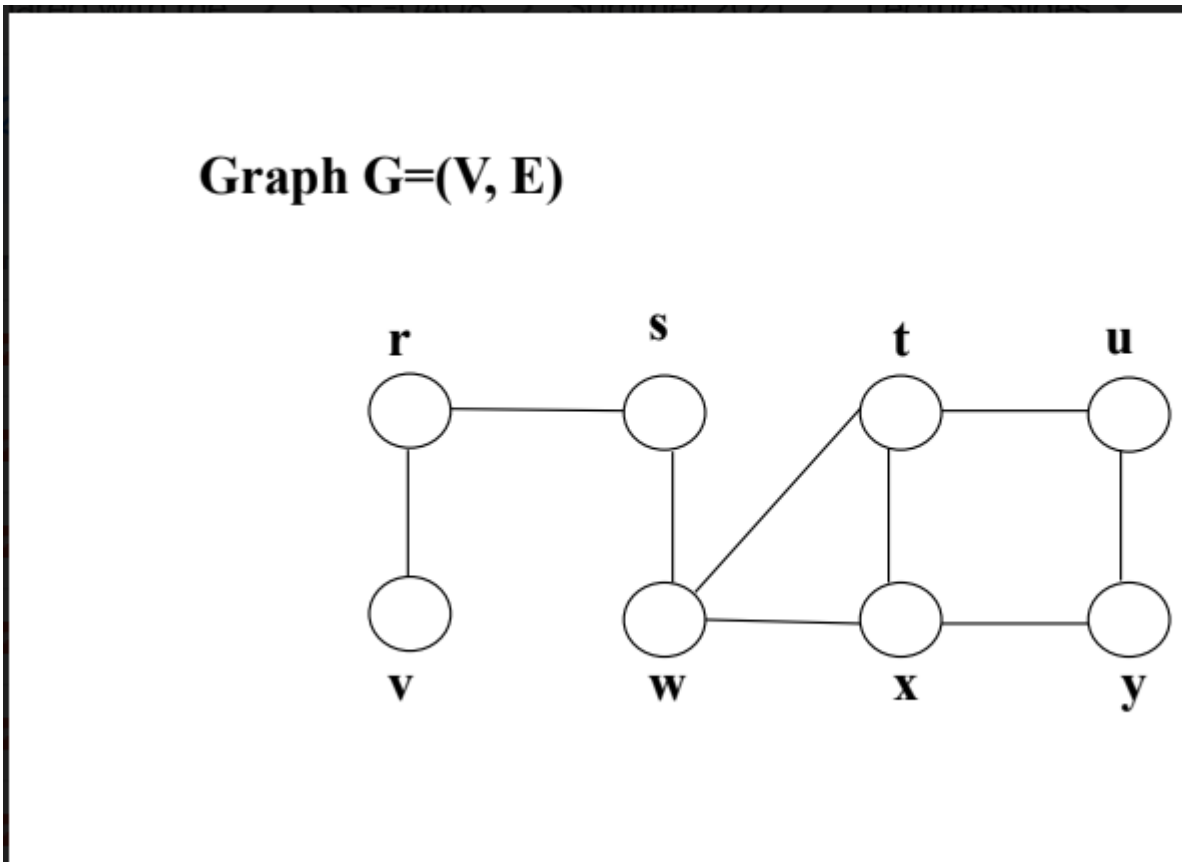

Fig. 1. Solving Method

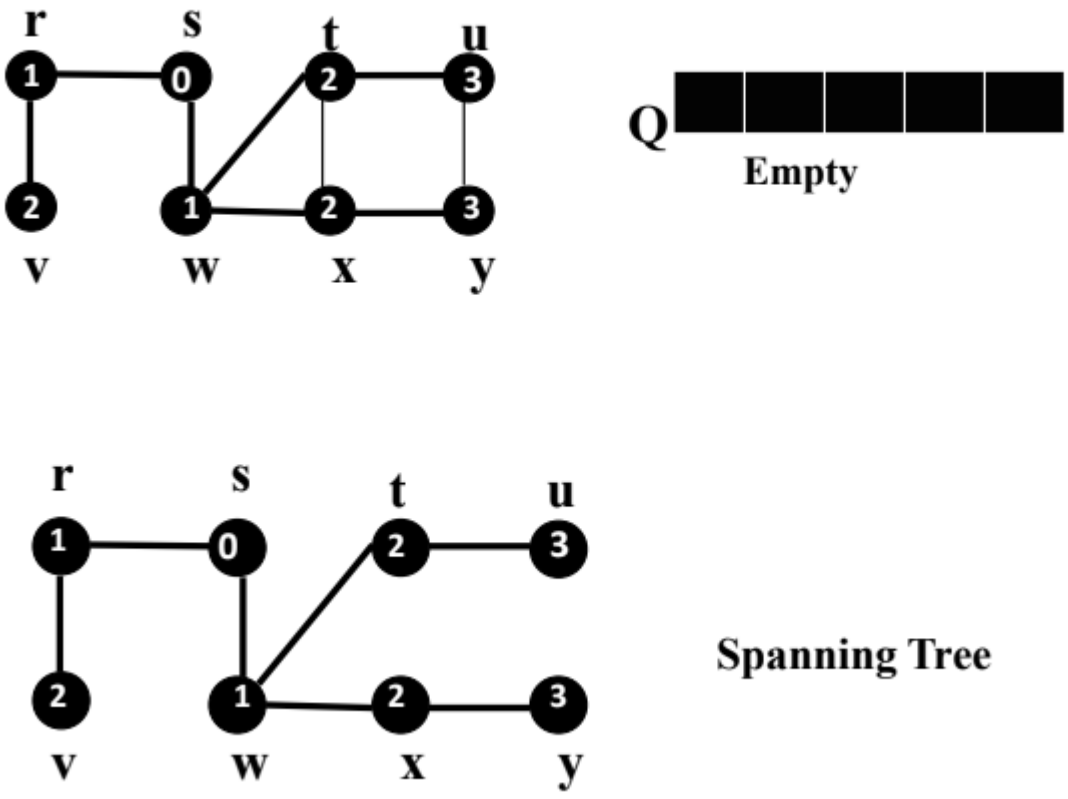

Fig. 2. Solving Method