"BREADTH FIRST TRAVERSAL TECHNIQUE USING TCL/TK"

Mini Project-I Report

Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

By

Aagam Jariwala (17BEC001)

Akshay Mishra (17BEC008)

Department of Electronics & Communication Engineering

Institute of Technology

NIRMA UNIVERSITY

Ahmedabad 382 481

November 2019

CERTIFICATE

This is to certify that the Mini Project Report entitled “Breadth First Traversal technique using TCL/TK” submitted by Aagam Jariwala (17BEC001) and Akshay Mishra (17BEC008) towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electronics & Communication Engineering of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this mini project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date: 20th November 2019

Name of Guide: Dr. Usha Mehta

Head of the Department
Department of Electronics & Communication Engineering
Institute of Technology, Nirma University
Ahmedabad

Undertaking for Originality of the Work (for all students)

We, Aagam Jariwala and Akshay Mishra, Roll No.17BEC001 & 17BEC008 respectively, give undertaking that the Mini Project entitled “Breadth First Traversal technique using TCL/TK” submitted by us, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in the field of Electronics & Communication Engineering of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere, it will result in severe disciplinary action.

____

Signature of Student

Date: _______________

Place: _______________

Endorsed by:

(Signature of Internal Guide)

Acknowledgement

I must acknowledge the strength, energy and patience that almighty **GOD** bestowed upon me to start & accomplish this work with the support of all concerned, a few of them I am trying to name hereunder.

I would like to express my deep sense of gratitude to my Supervisor, **Dr. Usha Mehta,** Professor, Electronics & Communication Engineering Department for her valuable guidance and motivation throughout my study.

I would like to express my sincere respect and profound gratitude to **Dr. Dhaval Pujara**, Professor & Head of Electronics & Communication Engineering Department for supporting and providing the facilities for my mini project work.

I would also like to thank all my friends who have helped me directly or indirectly for the completion of my project work.

No words are adequate to express my indebtedness to my parents and for their blessings and good wishes. To them I bow in the deepest reverence.

Aagam Jariwala
Akshay Mishra

Abstract

Breadth First Search (BFS) and other graph traversal techniques are widely used for measuring large unknown graphs, such as online social networks. It has been empirically observed that incomplete BFS is biased toward high degree nodes. In contrast to more studied sampling techniques, such as random walks, the bias of BFS has not been characterized to date.

Breadth-first search (BFS) is Associate in Nursing algorithmic program for traversing or looking out tree or graph knowledge structures. It starts at the tree root (or some capricious node of a graph, generally spoken as a 'search key'[1]), and explores all of the neighbor nodes at this depth before moving on to the nodes at subsequent depth level.

It uses the other strategy as depth-first search, that instead explores the node branch as way as attainable before being forced to turn back and expand different nodes.[2]

In this paper, we quantify the degree bias of BFS algorithm. To give a broader perspective, we took certain examples and implemented BFS algorithm to find the shortest path and discovering all possible paths along the way. Next, we simulated the BFS algorithm on MATLAB to find a navigation between obstacles to reach a destination point.

Index

LIST OF FIGURES

Chapter 1: Introduction

1.1 Introduction

Tool command language (TCL) is a powerful scripting language with programming features. It is available across Unix, Windows and Mac OS platforms. TCL is used for Web and desktop applications, networking, administration, testing, rapid prototyping, scripted applications and graphical user interfaces (GUI).

It is commonly used embedded into C applications, for rapid prototyping, scripted applications, GUIs, and testing. Tcl interpreters are available for many operating systems, allowing Tcl code to run on a wide variety of systems. Because Tcl is a very compact language, it is used on embedded systems platforms, both in its full form and in several other small-footprint versions.

Breadth-first search (**BFS**) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level. It uses the opposite strategy as depth-first search, which instead explores the highest-depth nodes first before being forced to backtrack and expand shallower nodes.

1.2 Motivation of the project:

Programming skill is the need of the hour and driven by the interest in that particular domain we decided to go for the language not so popular among teens these days - Tool command language. We were delighted to get an opportunity to implement the depth first search in the TCL. Artificial intelligence is one of the hot topics in recent times of which BFS is an integral part which made us more curious about the topic.

Our guide Dr. Usha Mehta saw this assignment as an opportunity to make students learn out of their department and expand their knowledge.

1.3 Objective of the project:

- To understand Breadth First Search algorithm.
- To gain in-depth knowledge of Tool Command Language.
- First step in basics of embedded systems.
- Implement Breadth First Search algorithm on Tool Command Language, MATLAB and Java.
- Using BFS for practical applications in navigation.

Chapter 2: Literature Review

In engineering, graph traversal (also referred to as graph search) refers to the method of visiting (checking and/or updating) every vertex in an exceedingly graph. Such traversals are classified by the order during which the vertices are visited. Tree traversal may be a special case of graph traversal. Unlike tree traversal, graph traversal might need that some vertices be visited more than once, since it's not essentially acknowledged before transitioning to a vertex that it's already been explored. As graphs become a lot of dense, this redundancy becomes a lot of current, inflicting computation time to increase; as graphs become a lot of distributed, the alternative holds true. Thus, it's sometimes necessary to recollect that vertices have already been explored by the algorithmic program, so vertices are revisited as sometimes as doable (or within the worst case, to forestall the traversal from continued indefinitely). this might be accomplished by associating every vertex of the graph with a "color" or "visitation" state throughout the traversal, that is then checked and updated because the algorithmic program visits every vertex. If the vertex has already been visited, it's neglected and therefore the path is pursued no further; otherwise, the algorithmic program checks/updates the vertex and continues down its current path. Several special cases of graphs imply the visitation of different vertices in their structure, and therefore don't need that visitation be expressly recorded throughout the traversal. a very important example of this can be a tree: throughout a traversal it should be assumed that each one "ancestor" vertices of the present vertex (and others counting on the algorithm) have already been visited. each the depth-first and breadth-first graph searches square measure diversifications of tree-based algorithms, distinguished primarily by the shortage of a structurally determined "root" vertex and therefore the addition of a knowledge structure to record the traversal's visitation state. A depth-first search (DFS) is an algorithmic program for traversing a finite graph. DFS visits the child vertices before visiting the sib vertices; that's, it traverses the depth of any explicit path before exploring its breadth. A stack (often the program's decision stack via recursion) is usually used once implementing the algorithmic program.

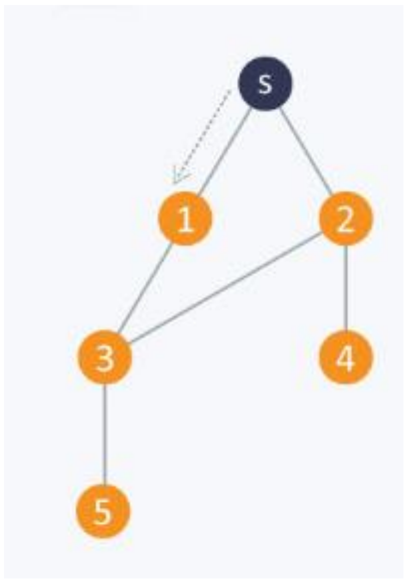
Chapter 3: Breadth First Traversal technique using TCL/TK

3.1 Pseudocode

```
procedure BFS(G,start_v):
    let Q be a queue
    label start_v as discovered
    Q.enqueue(start_v)
    while Q is not empty
        v = Q.dequeue()
        if v is the goal:
            return v
        for all edges from v to w in G.adjacentEdges(v) do
            if w is not labeled as discovered:
                label w as discovered
                w.parent = v
                Q.enqueue(w)
```

It uses a queue (First In First Out) instead of a stack and it checks whether a vertex has been discovered before enqueueing the vertex rather than delaying this check until the vertex is dequeued from the queue.[1]

3.2 Traversing Process



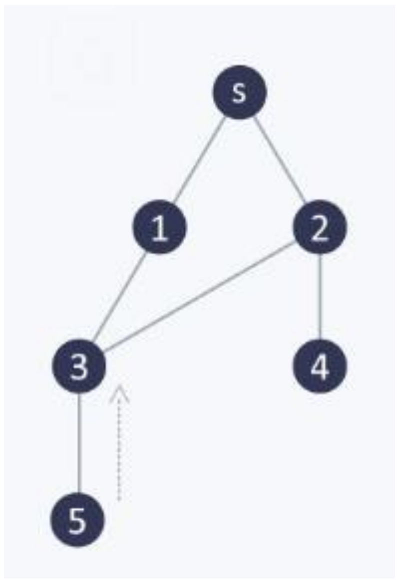


Figure 1 Traversing by taking s as root node

Figure 2 All the nodes have been traversed by using BFS

The traversing will start from the source node and push *s* in queue. *s* will be marked as 'visited'.[2]

First iteration

- s will be popped from the queue
- Neighbours of s i.e. 1 and 2 will be traversed
- 1 and 2, which have not been traversed earlier, are traversed. They will be:
    - Pushed in the queue
    - 1 and 2 will be marked as visited

Second iteration

- 1 is popped from the queue
- Neighbours of 1 i.e. s and 3 are traversed
- s is ignored because it is marked as 'visited'
- 3, which has not been traversed earlier, is traversed. It is:
    - Pushed in the queue
    - Marked as visited

Third iteration

- 2 is popped from the queue
- Neighbours of 2 i.e. s, 3, and 4 are traversed
- 3 and s are ignored because they are marked as 'visited'
- 4, which has not been traversed earlier, is traversed. It is:
    - Pushed in the queue
    - Marked as visited

Fourth iteration

- 3 is popped from the queue
- Neighbours of 3 i.e. 1, 2, and 5 are traversed
- 1 and 2 are ignored because they are marked as 'visited'
- 5, which has not been traversed earlier, is traversed. It is:
    - Pushed in the queue
    - Marked as visited

Fifth iteration

- 4 will be popped from the queue
- Neighbours of 4 i.e. 2 is traversed
- 2 is ignored because it is already marked as 'visited'

Sixth iteration

- 5 is popped from the queue
- Neighbours of 5 i.e. 3 is traversed
- 3 is ignored because it is already marked as 'visited'

The queue is empty and it comes out of the loop. All the nodes have been traversed by using BFS.

If all the edges in a graph are of the same weight, then BFS can also be used to find the minimum distance between the nodes in a graph.

3.3 Graph Traversal examples using TCL code

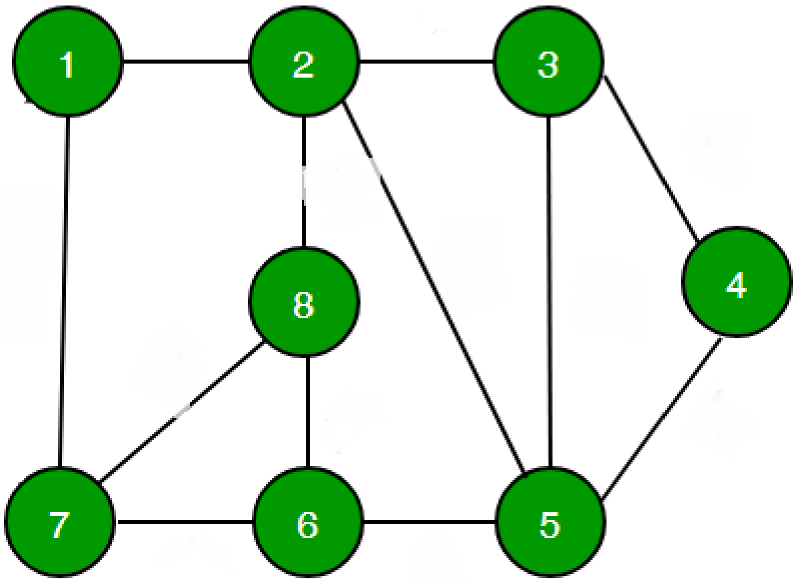
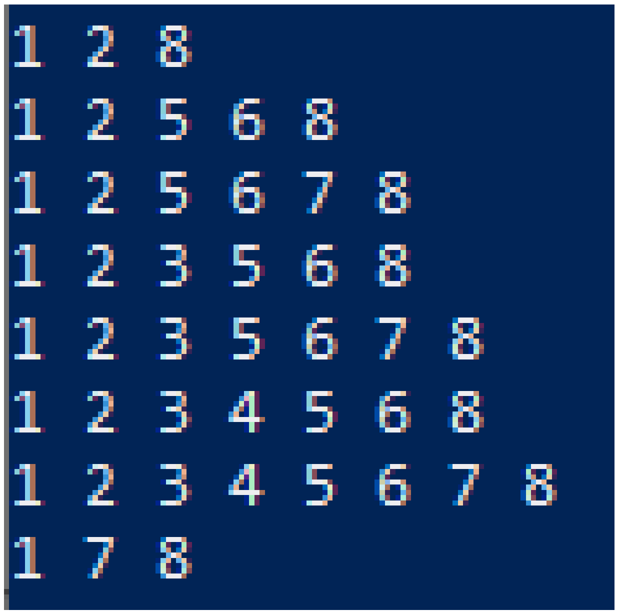


Figure 3 Graph Example 1

Start Node : 1 , End Node : 8



```
1 2 8
1 2 5 6 8
1 2 5 6 7 8
1 2 3 5 6 8
1 2 3 5 6 7 8
1 2 3 4 5 6 8
1 2 3 4 5 6 7 8
1 7 8
```

Figure 4 Example 1 Output

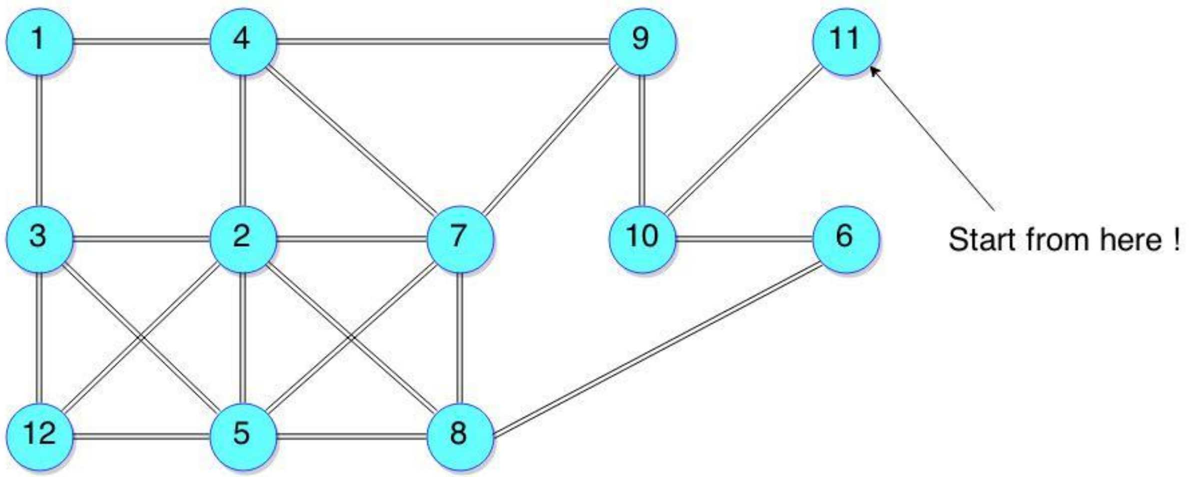
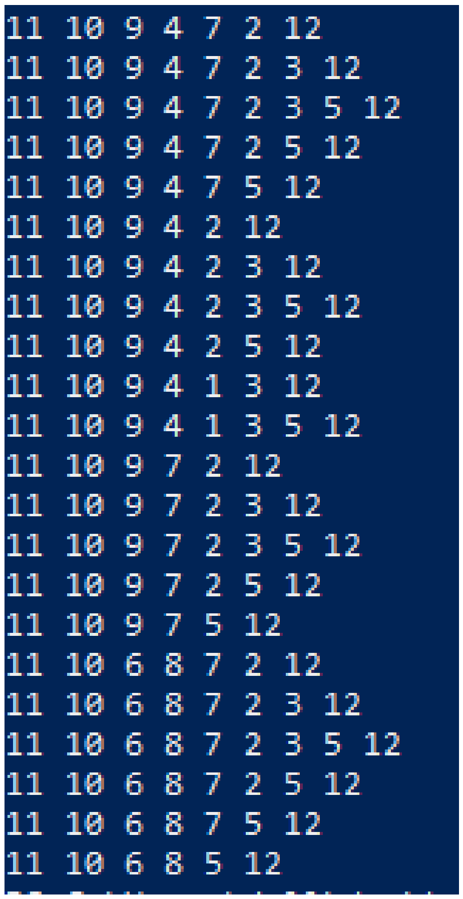


Figure 5 Graph Example 2

Start Node : 11 , End Node : 12



```
11 10 9 4 7 2 12
11 10 9 4 7 2 3 12
11 10 9 4 7 2 3 5 12
11 10 9 4 7 2 5 12
11 10 9 4 7 5 12
11 10 9 4 2 12
11 10 9 4 2 3 12
11 10 9 4 2 3 5 12
11 10 9 4 2 5 12
11 10 9 4 1 3 12
11 10 9 4 1 3 5 12
11 10 9 7 2 12
11 10 9 7 2 3 12
11 10 9 7 2 3 5 12
11 10 9 7 2 5 12
11 10 9 7 5 12
11 10 6 8 7 2 12
11 10 6 8 7 2 3 12
11 10 6 8 7 2 3 5 12
11 10 6 8 7 2 5 12
11 10 6 8 7 5 12
11 10 6 8 5 12
```

Figure 6 Example 2 Output

4. MATLAB Simulation

Shown below is the application of BFS in navigation.[3]

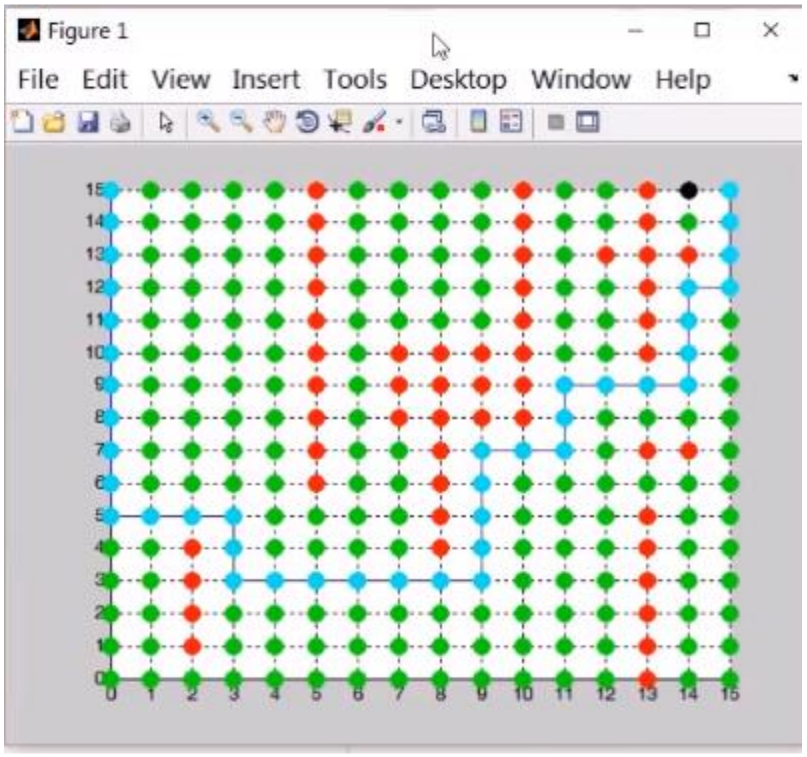


Figure 7 MATLAB Simulation of BFS

5. Conclusion

Through this project we implemented the Breadth First Search Algorithmin Tool Command Language and MATLAB.We compared Breadth First Search with Depth First Search and saw how they both have their different uses. We saw two important applications of Breadth First Search i.e. Graph traversal scheme and Navigation and implemented in TCL and MATLAB respectively..

6. References

[1]. https://en.wikipedia.org/wiki/Breadth-first_search

[2].http://opendatastructures.org/versions/edition0.1e/ojava/12_3_Graph_Traversal.html#SECTION001531000000000000000

[3] https://www.youtube.com/watch?v=Bcwh7jCimPg

[4]. M. Kurant, A. Markopoulou and P. Thiran, "On the bias of BFS (Breadth First Search)," *2010 22nd International Teletraffic Congress (lTC 22)*, Amsterdam, 2010,

[5] S. Beamer, K. Asanovic and D. Patterson, "Direction-optimizing Breadth-First Search," *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Salt Lake City, UT, 2012