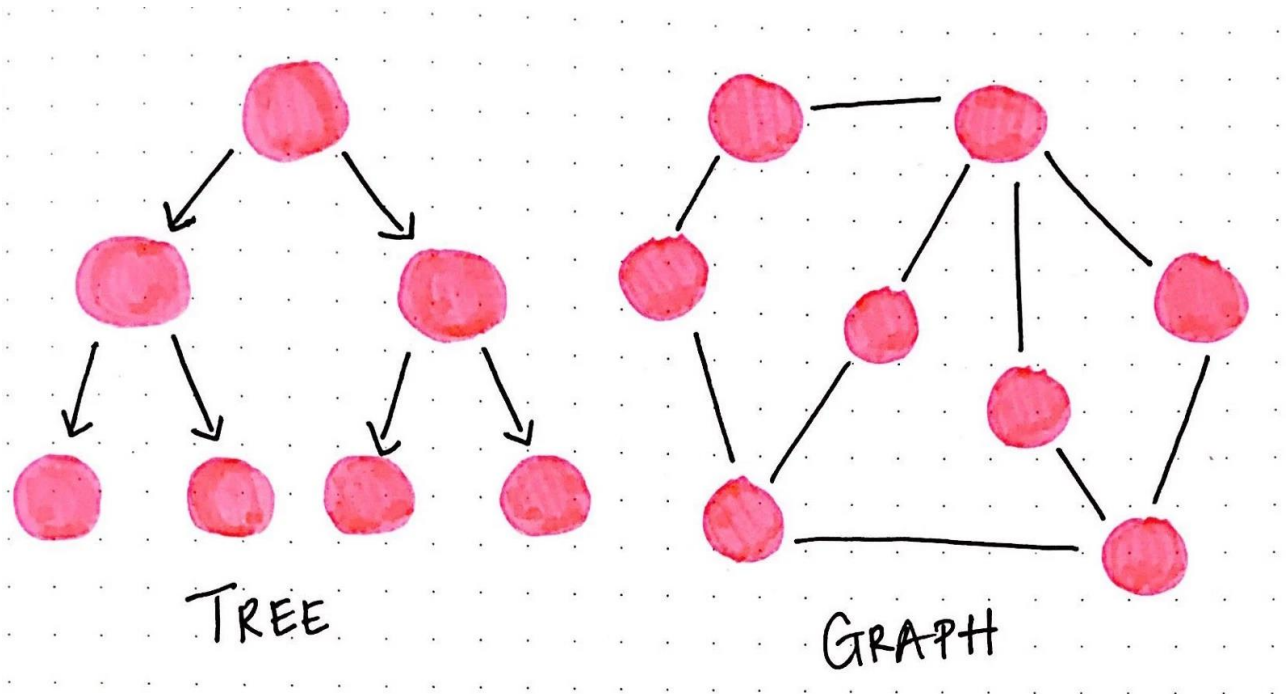


Graphs, Trees, Containers and Recursion



Semester 3 SD

6_Graphs-and-c++containers

Jan 23, 2023.

Student: Andre Sanao

Course: Technology

Table of Contents

Introduction.....	4
Assignment 1 (recursion):	5
Assignment 2 (graphs - shortest path):	7
Conclusion	8
Reference	8

Acronyms

<i>Acronym</i>	<i>Meaning</i>
<i>ICT</i>	→ Information and communication technologies
<i>BFS</i>	→ <i>Breadth-first Search</i>
<i>STL</i>	→ <i>Standard Template Library</i>

Table 1 - List of acronyms used throughout the report

Introduction

The assignment on which this document presents an overview of various types of coding methods used to represent data structures and algorithm. C++ is a vast language that consist of STL that can be used for implementing into devices or products and are used to perform a specific function or set of functions. They are commonly used in technology. The following section will provide data structure and algorithm used to represent technology and provide a report of how they can be implemented.

Assignment 1 (recursion):

In this part of the assignment, we are going to find the depth of a binary tree. A binary tree is a popular data structure that is non-linear in nature. Think of it a like a real tree that extends to branches of the tree. (See figure 1)

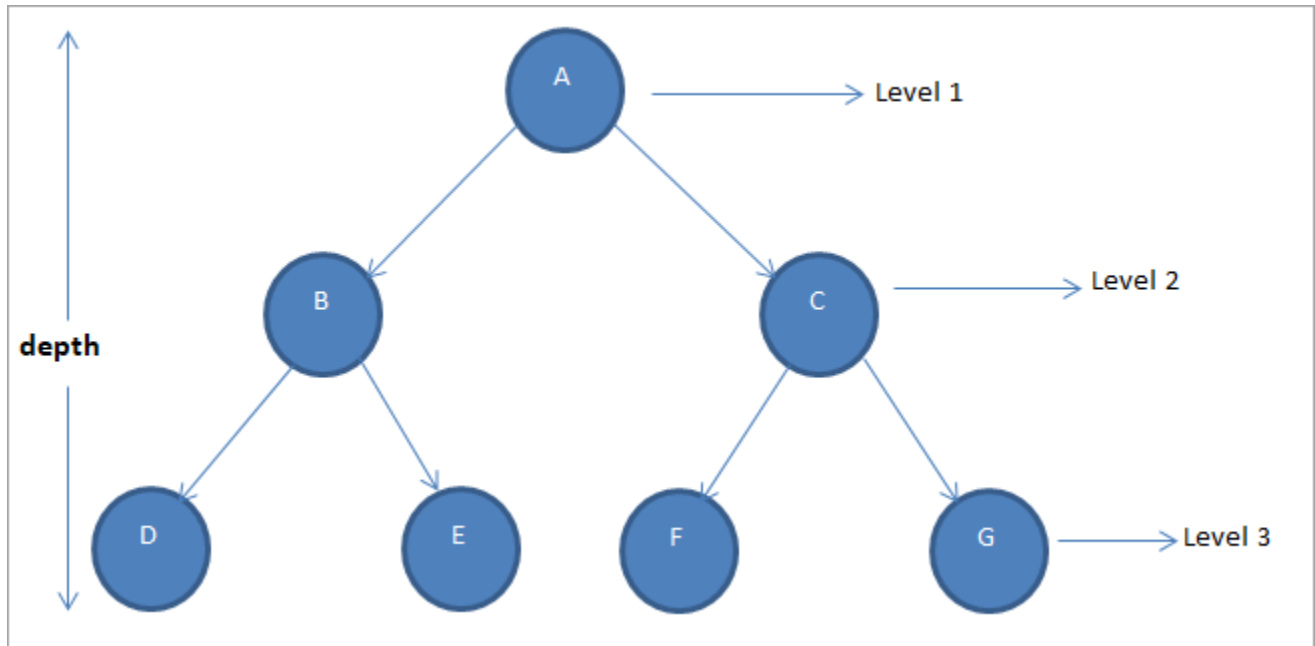


Figure 1 Binary Tree structure

Unlike the previous assignments which are linear in nature, a binary tree represents a hierarchical structure.

This assignment is also required to implement recursion in the function. Recursion is where the function keeps calling itself until for example an if-statement breaks the loop.

For this assignment, I created a Node and Tree class that to find the depth of a binary tree (See figure 2).

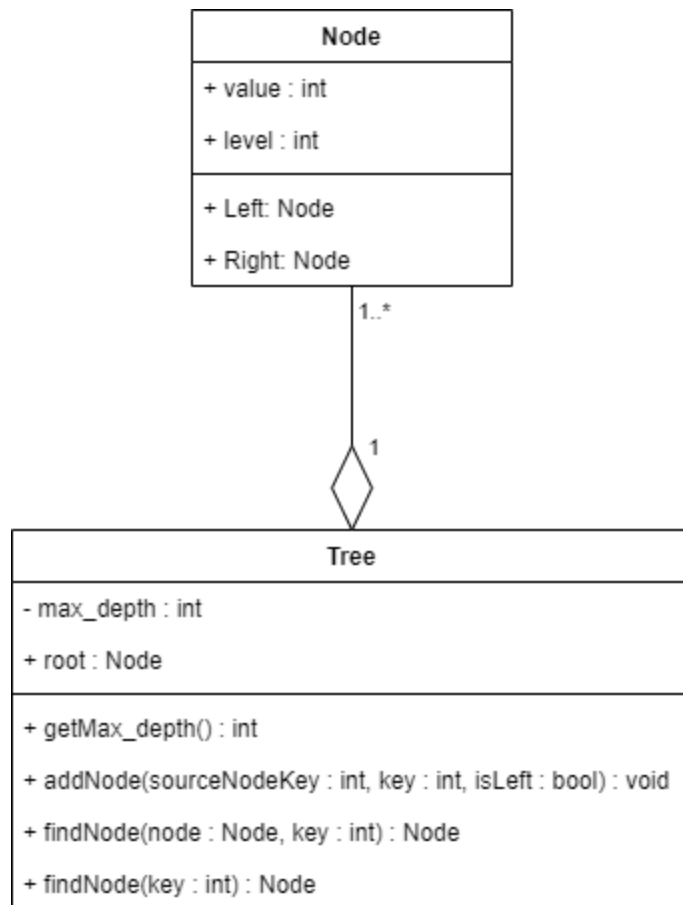


Figure 2 Part 1 UML diagram

Node class is used as pointers for the left and right child of the parent node.

Tree class will add nodes and find the depth of the tree with N elements.

I researched and implemented this searching algorithm because it is basic and easy to understand.

Assignment 2 (graphs - shortest path):

For this part of the assignment, we have a network of N nodes numbered from 1 to N. These networks are connected bidirectional. The task for this assignment is to find the shortest path using graphs algorithm. (See figure 3)

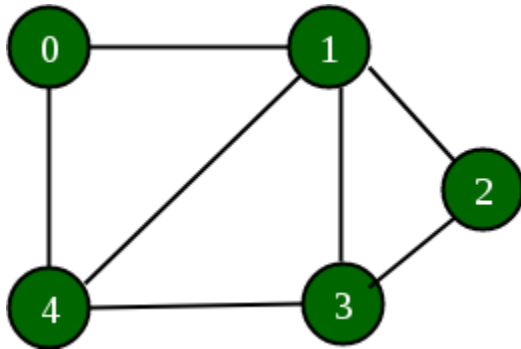


Figure 3 Graphs representation

Queue
- queue : vector<int>
+ push(x : int) : void
+ pop() : void
+ front() : int
+ empty() : bool

Figure 4 Part 2 UML diagram

For this algorithm I used a Queue class (See figure 4). I used the provided link from the assignment as a start to research what BFS is. With that as a start I kept on doing research tried different methods to implement BFS with queue. I started using plain queue class with pointers but it wasn't giving me the results that I wanted. So I tried using vectors to implement graphs using adjacency list representation. The idea of this method is to represent the graph as an array of vectors. Every vector will represent the adjacency list of a vertex

Conclusion

This assignment gave me learning outcomes of implementing recursion and learning 2 new algorithms. This assignment was different from the previous assignment because we are basically not locked anymore from using libraries. I tried using smart pointers for the first part but failed and I think I need more time to be able to understand and implement it correctly so I can avoid memory leaks in the future. For part 2, I had to read the article a few times to understand BFS and implement it in the code. This assignment thought me more about data structures and algorithm that can be used in the future and hopefully learn more data structures so we can make more optimized codes.

Reference

GeeksforGeeks. (2023, January 11). *Introduction to binary tree - data structure and algorithm tutorials*. GeeksforGeeks. Retrieved January 23, 2023, from <https://www.geeksforgeeks.org/introduction-to-binary-tree-data-structure-and-algorithm-tutorials/>

Joshi, V. (2017, September 18). *Going broad in a graph: BFS Traversal*. Medium. Retrieved January 23, 2023, from <https://medium.com/basecs/going-broad-in-a-graph-bfs-traversal-959bd1a09255>

GeeksforGeeks. (2023, January 18). *Graph and its representations*. GeeksforGeeks. Retrieved January 23, 2023, from <https://www.geeksforgeeks.org/graph-and-its-representations/>