Georgios Ioannou

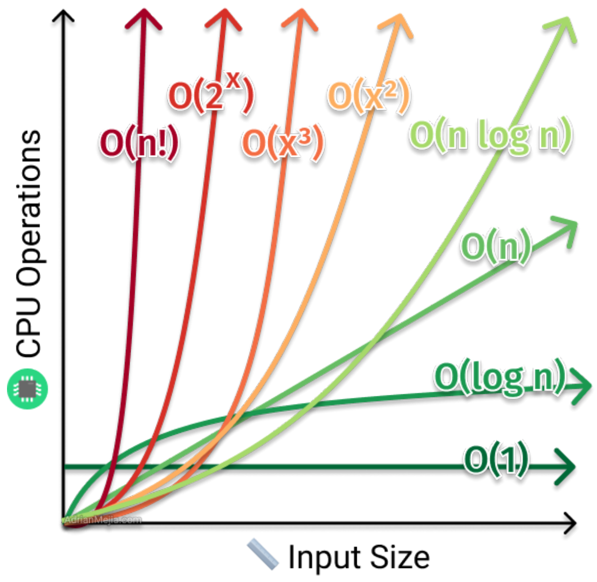
CSC 44800

06 November 2023

THE CITY COLLEGE OF NEW YORK

DFS and BFS Time and Space Complexity:

HW 2-2



Georgios Ioannou

CSC 44800

06 November 2023

Georgios Ioannou

CSC 44800

06 November 2023

TABLE OF CONTENTS

1. Introduction 3

2. Time Complexity 3

3. Space Complexity 4

4. DFS And BFS Time and Space Complexity 4

5. Conclusion 5

Georgios Ioannou

CSC 44800

06 November 2023

DFS and BFS Time and Space Complexity:

HW 2-2

INTRODUCTION

In this report we will go over what time and space complexity are and then discuss the time and space complexity for both Depth-First Search (DFS) and Breadth-First Search (BFS). We will utilize the two notebooks that implemented DFS and BFS and can be found in this repository.

TIME COMPLEXITY

Time complexity is a fundamental concept in computer science and algorithm analysis that quantifies the computational efficiency of an algorithm or program. Time complexity measures how the execution time of an algorithm grows with the input size. This provides valuable insights into the algorithm's scalability and performance. Time complexity is often expressed using the big O notation, which describes the upper bound (worst case) on the number of basic operations an algorithm performs in relation to its input. Time complexity is essential for designing and selecting algorithms that can solve problems efficiently, enabling developers to make informed choices about which algorithm to use in various computational tasks and ensuring that software systems can handle increasingly larger datasets without significant performance degradation. Even though nowadays we have powerful machines it is always best to design algorithms that are efficient, optimize, and with low time and space complexity

Georgios Ioannou

CSC 44800

06 November 2023

SPACE COMPLEXITY

Space complexity is yet another crucial concept in computer science and algorithm analysis. Space complexity focuses on the amount of memory or storage an algorithm requires to execute, and how this usage scales with input size. Space complexity measures the efficient utilization of memory resources during an algorithm's operation, and like time complexity, it's often expressed using the big O notation. Analyzing space complexity is essential for designing memory-efficient software, especially when working with large datasets. Balancing time and space complexity is a central consideration in algorithm design, as minimizing memory usage can lead to more efficient and responsive applications, while excessive memory consumption can lead to issues like slowdowns or even system crashes.

DFS AND BFS TIME AND SPACE COMPLEXITY

Both DFS and BFS have a time complexity of O(V+E), where V is the number of vertices and E is the number of edges. This is because in the worst case scenario we need to visit all vertices and edges.

The iterative approach of DFS using a stack that can also be found in this repository' Jupyter notebook, has a space complexity of O(V) because the stack is used to keep track of vertices to visit, and in the worst case, it can store all the vertices. The space complexity of BFS is determined by the queue used to keep track of the visited vertices during the traversal of the graph/tree. In the worst case, the queue can store all the vertices at the same time, so the space complexity is O(V).

Georgios Ioannou

CSC 44800

06 November 2023

CONCLUSION

Overall, the iterative implementation of DFS that uses a stack and the implementation of BFS that uses a queue both have a time complexity of O(V+E) and a space complexity of O(V), where V is the number of vertices and E is the number of edges.