Algorithm Design and Analysis

FINAL PROJECT PROPOSAL

Comparison Of

Four Weird Algorithms

Prepared by:

Baal's Abandoned Children

(Jocelin, Nicholas & Tiffany)

Odd semester 2nd year 2022

Computer Science Program

Binus University International

Table of Contents

Project name: Comparison Of Algorithms

Group members:

1. Tiffany Widjaja - 2502059360 (L3AC)

2. Nicholas Valerianus Budiman - 2502055596 (L3AC)

3. Jocelin Wilson - 2501963330 (L3AC)

Background:

We chose to do a comparison of algorithms for our final project so that we can study the difference between their efficiency such as the run time, memory consumption, and finding out in what situation would it be most optimal to use them. The three algorithms we will be comparing are all sorting algorithms however there is a twist to it as they are not the most popular algorithms. In fact, these are the strangest sorting algorithms that we didn't know existed. The Four Sorting algorithms are:

- Spaghetti
- Stalin
- Sleep
- Cocktail

Spaghetti Algorithm is one of the weirdest we've encountered and that is because it involves a physical aspect to it since generally speaking; algorithms tend to be based on abstract logic sequences. This is not the case for the Spaghetti sorting algorithm as it can only sort positive real numbers and is not meant to be a practical sorting algorithm. Only for this specific algorithm, there will be no code provided however we will still compare this algorithm with the rest. Just

not comparing it from it's memory usage or run time, perhaps only its time complexity. That being said, the spaghetti sort has an O(n) time complexity as its best case with O(n) time complexity being its worst case. This leads to an average of an O(n) time complexity overall.

How the spaghetti algorithm works is, for every individual value that you want to sort, we need to cut a piece of spaghetti into the length that is equal to the value you want to sort. This cutting process takes O(1) time for each value and a total of O(n) or all n values. Afterwards, take these pieces of spaghetti in your hand and lay each piece on a table so that all the pieces are lined up at one end vertically. The tallest piece is determined to be the largest value, and this process is continuous so the next tallest piece is then the next largest, etc.

The stalin sorting algorithm is a sorting algorithm that sorts arrays and numbers by eliminating numbers that are not positioned correctly. This occurs repeatedly until the numbers are sorted in order. For example, we have an unsorted array: {1,2,4,3,5,8,7,10,9,13,11,12}. After implementing stalin sort, the array turns to {1,2,4,5,8,10,13}. As you can see it eliminates all the numbers that are not placed in order.

Sleeping sort, also known as the king of laziness, is a sorting algorithm that involves time. So the element having the least amount of sleeping time wakes up first and the number gets printed and then the second least element and so on. The largest element wakes up after a long time and then the element gets printed at the end. Due to the elements being dependent with the time, this doesn't allow negative numbers in the array. The downside of this is that this algorithm doesn't produce a correct sorted output every time. This generally happens when there is a very small number to the left of a very large number in the input array.

Finally, there is the cocktail algorithm, which is another variation of the bubble sort. The algorithm gets its name from the way bubbles sort by repeatedly passing through the array, comparing adjacent elements and swapping them if it is not sorted which is similar to the way a bartender mixes cocktails by shaking them. The first stage is traversing the array starting from the left all the way to the right, just like Bubble Sort. During the loop, adjacent items are compared and if the value on the left is greater than the value on the right, then values are swapped. At the end of the first iteration, the largest number will reside at the end of the array. The second stage goes through the array in the opposite direction; starting from the item just before the most recently sorted item, it goes back to the start of the array. Here also, adjacent items are compared and are swapped if required. Similar to bubble sort, the worst and average case complexity of cocktail sort is O(n2), while its best case is O(n) time complexity. When it comes to speed, cocktail sort is around two times faster than bubble sort since it will only require two traversals meanwhile traverses back and forth while sorting the first element and the last element which it then proceeds to sort until it reaches the middle and everything is sorted. This algorithm is also known as the happy hour sort, ripple sort, and shuttle sort.

Problem

There are plenty of algorithms available for programmers to choose from however not everyone knows which algorithm is suitable to implement in some case scenarios. Factors such as efficiency like memory usage, time complexity, run time, and its simplicity is undeniably important for us to consider before deciding on which algorithm to use to solve a problem. No one would like their program to be slow or take an unnecessary amount of memory. Our group would like to test out the advantages and disadvantages between the four strangest algorithms

for different problems, operations and data sizes. Therefore we are doing a comparison between the Spaghetti, Stalin, Sleep, and Cocktail method. This has to be done to make sure that the most effective and efficient algorithm can be used on the problems for good optimization and efficient time and memory usage. In this project we will compare and study the algorithms run time, memory usage, pros, cons, and time complexity so that other fellow programmers and ourselves included can have a good idea of what algorithms to use on different scenarios for optimal solutions in the future.

What is to be measured:

Since we are comparing algorithms we would have to measure run time, memory consumed, effort taken to apply each algorithm, and see which works best in a certain situation and find where they are optimal at. We will be using 8 different sizes to measure which algorithm is most suited for sorting which array. These cases of array will consist of a nearly sorted, random and reverse sorted numbers. The arrays size would be 1000, 10000, 30000, 50000, 100000, 300000, 600000, 1000000.

Proposed Solution:

We propose to compare three weird different algorithms and have it coded using C++. We believe that coding these sorting algorithms in C++ would help us in lots of ways as it allows us to easily create structures and allocate memory to certain objects. C++ is more preferred for coding in algorithms anyways if we were to look for all-around efficiency as it has the best data structures out of three in comparison with Python and Java. Some of the best algorithms are

ready to use in C++ from the standard template library and thus be more convenient for us if we ever decide to compare more algorithms or change into another. For the algorithms, we plan to compare four which are the Spaghetti, Stalin, Sleep, and Cocktail method. By implementing these different sorting algorithms and comparing them, we hope to finally reach a conclusion and have a better understanding on when to apply certain algorithms in the future. Such proposed solutions are still subject to change and are just part of our brainstorming during our discussion.

The Flow:

1. Prepare the code algorithm.
2. Prepare three 8 different sizes of of nearly sorted, random and reversely sorted arrays.
3. Find the time and space complexity of algorithms.
4. Comparison between memory, run time, and code simplicity of the algorithms.
5. Identifying pros and cons of each algorithm
6. Identifying real life samples of when such algorithm is used
7. We conclude which is the best situation and worst situation where the algorithm is applied to the several cases mentioned.

Findings:

Conclusion & Recommendations:

Resources

https://www.geeksforgeeks.org/cocktail-sort/

https://stackoverflow.com/questions/12324822/sleep-sort-in-c-11

https://www.geeksforgeeks.org/sleep-sort-king-laziness-sorting-sleeping/