**Project 1 – Matrix Manipulation and Sorting**

Presented to: Professor Phillip Servio

Prepared By:

Bizhan Alatif (260907005) – bizhan.alatif@mail.mcgill.ca

Ngan Jennifer Tram Su (260923530) – jennifer.tramsu@mail.mcgill.ca

CHEE 390 – Computational Methods in Chemical Engineering

Department of Chemical Engineering

McGill University

Fall 2021

Table of Contents

[1. Objective 3](#_Toc83215699)

[2. Flowchart 4](#_Toc83215700)

[3. Results and Discussion 5](#_Toc83215701)

[3.1 Pivot Selection 5](#_Toc83215702)

[3.2 In-Place Sorting 5](#_Toc83215703)

[3.3 Arithmetic Overflow 6](#_Toc83215704)

[4. Conclusion 7](#_Toc83215705)

[5. References 8](#_Toc83215706)

# 1. Objective

The objective of this report was to implement a program that sorts a 3D matrix in two steps: (1) sort each page of the matrix in ascending order of their trace (sum of diagonals), and (2) apply quicksort to sort each individual plane.

* ??

# 2. Flowchart

Figure 1 follows the path of an input 3D matrix through the implemented sorting algorithm.

Diagram

Description automatically generated

Figure 1 Quicksort Algorithm Flowchart

# 3. Results and Discussion

## 3.1 Pivot Selection

In sorting algorithms, time complexity is poo

## 3.2 In-Place Sorting

When creating an array in MATLAB, the assigned array is stored in a unique block of memory, with a separate block of memory holding information describing the array (e.g. dimensions). When a copy of an array is created, no new block of memory is made; however, once changes to this copy are made, the information of this new modified array are stored in a separate memory block, with an additional memory block storing the array’s information. As a result of re-assigning previously created arrays and, subsequently, modifying them, more blocks of memory are used up leading to longer computing times.

In the case of sortingz, creating array copies which are sorted separately from the parent array may lead to the worst-case scenario in space complexity (i.e. the amount of memory used in an algorithm). If all available blocks of memory are filled while running this algorithm,

## 3.3 Arithmetic Overflow

Computers represent information with binary digits, otherwise known as bits. Having a base of 2, these binary digits can be converted to base-10 to represent integers. However, limitations in memory prevent computers from fully expressing every value in the integer domain. These limitations are commonly bounded by 32 or 64 bits, depending on the computer. For example, the largest positive (negative) value that can be expressed with 32 bits is 2,147,483,647 (-2,147,483,648) (1). Any attempt to express a larger numeric value will result in arithmetic overflow. Simply put, arithmetic overflow occurs when a computer attempts to store a numeric value that exceeds the bounds of its memory.

In the case of quicksort, arithmetic overflow may occur during pivot selection. A common pivot choice is the middle index of the array, whose value can be obtained by equation (1).

|  |  |
| --- | --- |
|  | (1) |

However, should the matrix be sufficiently large, the sum operation may result in a numeric value whose binary representation exceeds the computer’s memory, leading to arithmetic overflow. Consequently, (assuming only positive values) the program will return a negative index and throw an error as MATLAB does not support negative indexing. Ideally, this must be avoided so that the quicksort algorithm can be generalized for an input 3D matrix of any size. Certain pivot selections can avoid this problem. Arithmetic overflow never occurs when adding two numbers of different sign. Therefore, pivot selection methods that follow this property may be chosen to safeguard against arithmetic overflow (e.g. random selection).

# 4. Conclusion

Of the available sorting methods, the algorithm presented in this report follows the quicksort method. With a time complexity of , quicksort is a fast divide-and-conquer method that outperforms many others. Selecting a pivot, this method compares values on either side of it, swapping them until the end condition is met and recursively calls the method on the two resulting subarrays. To avoid the worst-case scenario, the implementation presented in this report uses a random pivot selection; this also avoids the case of arithmetic overflow as integers of the same sign are never added.

- more stuff spae compledxity idk

# 5. References

[1] IBM Corporation. (n.d.). *Signed and Unsigned Integers*. IBM. Retrieved September 21, 2021, from https://www.ibm.com/docs/en/aix/7.1?topic=types-signed-unsigned-integers