SEMESTER 1, 2021/2022

CSCI 3302 DATA STRUCTURES AND ALGORITHMS II

Section 04

LIBRARY SYSTEM

PREPARED BY: GROUP G

NAME	MATRIC NO.
MOHAMED MOUBARAK MOHAMED MISBAHOU MKOUBOI	1820705
OBEID SALEM AHMED BASHWAR	1813179
WARSAME ABDI YUSUF AHMED	1724685

LECTURER:

Dr.ASMARANI BINTI AHMAD PUZI

Table of contents

Introduction

The old notion of a library as a repository of information is fading, giving birth to quickly evolving concepts such as digital libraries, electronic libraries, virtual libraries, wireless libraries, and even a paperless society. Digital libraries allow users to access desired information by emphasizing electronic objects or digital form rather than traditional printed book form and thus provide universal access to information distributed on networks, at the click of a mouse button on your personal computer connected to the networks. Due to the facts mentioned before, we came up with a major goal which is the library system, which will employ a sorting algorithm to assist students in swiftly and easily sorting books based on their Title, Author, or Date of Publication. We will build the method in C++ language, which for almost forty years, has been used for game development, software engineering, data structures, designing browsers, operating systems, apps, and other purposes to address real-world problems like the one we are discussing.

Real Problem-Solving Questions

- IIs this program/algorithm helps individuals and society who work/study in education
- How sorting algorithm helps the society to gain more education
- why did you choose a sorting algorithm other than other algorithms
- Describe your program and how will help society in Humanising Education

Why this problem

Arranging books in a library is a major duty that takes a lot of time and work. As a result, we devised the notion of sorting books using a merge algorithm, which is a sorting algorithm that receives an array as input, performs specified operations on the array (also known as a list), and returns a sorted array. We will assist the library management in sorting and arranging the books in a timely and accurate manner. This project can assist the library system in having a Reliability of Information, which is the content that has been noted that the majority of the material posted on various websites and social media platforms such as blogs, twits, wikis, Facebook, and so on is not legitimate and is also available in digital formats. Furthermore, it can aid in Data Security, which is all about keeping data secure, and there are several major threats to data saved on digital media, such as system crashes, damaged discs, power outages, accidentally deleting or overwriting files, computer viruses, hackers, natural calamities, money-making, revenge, and so on.

The Algorithm used

We used the sorting algorithm to sort the books by year of publication, author name, or book title

Explanation of how it works

Calculation

Time Complexity of Merge sort for the Library System
In the worst case, in every iteration, we are dividing the book array into further 2 arrays. This will result in a n log n process and this has to be done for n number of times leading to n log n time in total executing the sorting of books. Hence, the time complexity in the worst-case scenario is O(n log n).

Space Complexity of Merge sort
We need an additional size since we are copying the array into subLeft and subRight.

Pseudocode

```
Start
Initialize array of books (arr)
Print book with no sorting

Print book sorted by book title
Call MergeSort(arr, left, right, by boo title)

Print book sorted by author’s name
Call MergeSort(arr, left, right, by author name)

Print book sorted by year of publish
Call MergeSort(arr, left, right, by year)

merge(arr, left, mid, right, typeOfSort):
        merge the smaller books into a new book list by the type of sort (title or author or
year)

mergeSort(arr, left, right, typeOfSort):
        if left > right
                Return
        mid = (left+right)/2
        Call mergeSort(arr, left, mid)            divide furthermore.
        Call mergeSort(arr, mid+1, right)         divide furthermore.
        Call merge(arr, left, mid, right, typeOfSort)
```

end

Real Program

```
/*
This program is written in C++

This program sorts books based on their Title, Autor or date of publish
Using Merge Sort Algorithm
*/

#include <iostream>
using namespace std;

// Book structure
struct Book{
	string bookTitle;
	string author;
	int yearOfPublish;
};

int typeOfSort = 0; //1 = sort by book title,	2 = sort by Author name,	3 = sort by date of
publish

//function Protoypes
void merge(Book [], int, int, int);
void mergeSort(Book [], int, int);
void printArray(Book [], int);


int main()
{
	//initilize the books of array
	int n = 4;
	Book array[n];

	// Details of Book 1
	array[0].bookTitle = "Introduction to Algorithms";
	array[0].author = "Thomas	";
	array[0].yearOfPublish = 2022;

	// Details of Book 2
	array[1].bookTitle = "Grokking Algorithms	";
	array[1].author = "Aditya Bhargava";
	array[1].yearOfPublish = 2016;

	// Details of Book 3
```

```
        array[2].bookTitle = "The Algorithm Design Manual";
        array[2].author = "Steven       ";
        array[2].yearOfPublish = 1999;

        // Details of Book 4
        array[3].bookTitle = "Advanced Data Structures";
        array[3].author = "Peter Brass";
        array[3].yearOfPublish = 2002;

        int arraySize = sizeof(array) / sizeof(array[0]); // save the size of the array to be able
to divide them

        cout << " ********************************************************************* \n";
        cout << " This program sort books based on it's Title, Autor or date of publesh \n
using Merge Sort Algorithm \n";
        cout << "\n ********************************************************************* \n";

        cout << "\n\n Raw data, No sorting applied on books:\n\n";
        printArray(array, arraySize);

        do {
                cout <<" How would you like to sort the books?"
                <<"\n 1 - Sort based on the book title"
                <<"\n 2 - Sort based on the author name"
                <<"\n 3 - Sort based on year of publish\n ";
                cin >> typeOfSort;
        } while (typeOfSort<1 || typeOfSort>3);

        mergeSort(array, 0, arraySize - 1); //fuction call to sort the books
        printArray(array, arraySize);

        return 0;
}

void printArray(Book array[], int arraySize){
        cout <<"\n Order       Book Title                         Aurhor          Publish Date"

<<"\n______________________________________________________________________"
<<endl;

        for (int i = 0; i < arraySize; i++)
                cout <<          "  " <<i+1 << "  "
                        << array[i].bookTitle << "        "
                        << array[i].author <<"  "
                        << array[i].yearOfPublish<<endl;
        cout <<endl;
}
```

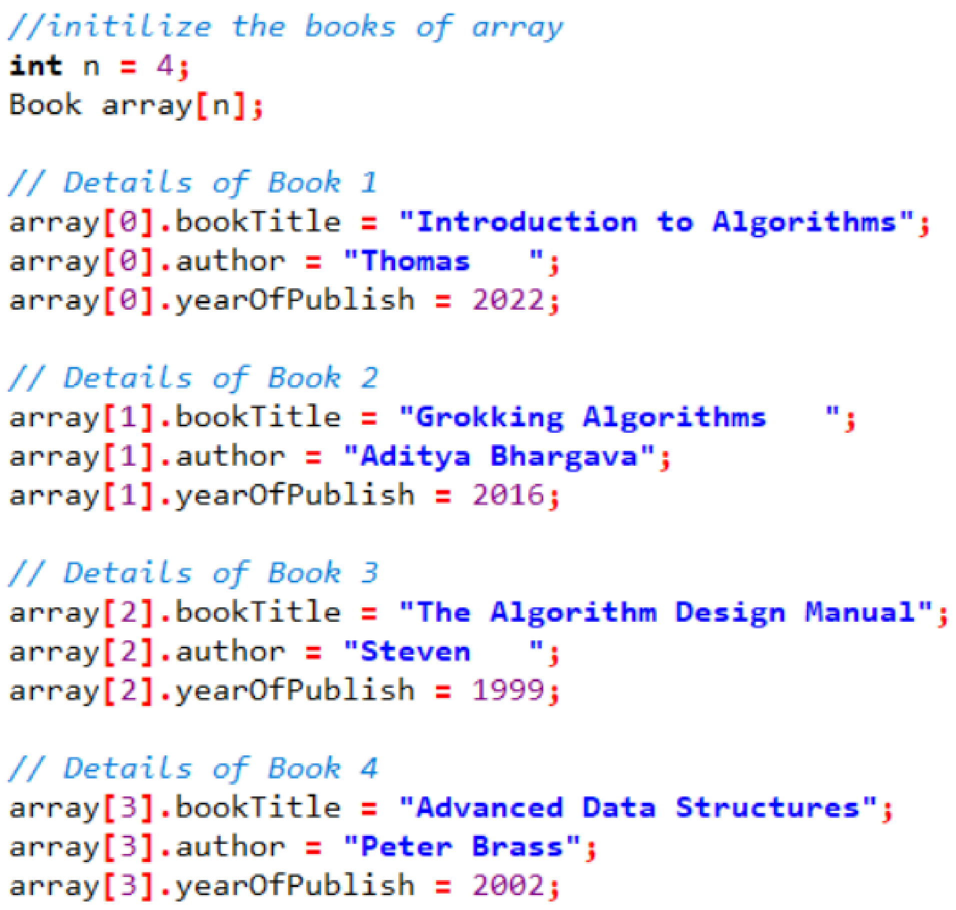
```
void mergeSort(Book array[], int const start, int const end){
	if (start >= end){ //no more diviton could be performed
		return;
	}

	int mid;
	mid = start + (end - start) / 2; //to caculate the middle
	mergeSort(array, start, mid);  //divid recursivly the first half
	mergeSort(array, mid + 1, end);	//divid recursivly the second half
	merge(array, start, mid, end); // finally conqer the dividens by marging them in
acending order
}


void merge(Book array[], int const left, int const mid, int const right){
	int subLeft = mid - left + 1;
	int subRight = right - mid;

	// temp arrays
	Book *leftArray = new Book[subLeft];
	Book *rightArray = new Book[subRight];

	// Copy data to temp arrays leftArray[] and rightArray[]
	for (int i = 0; i < subLeft; i++){
		leftArray[i] = array[left + i];
	}

	for (int j = 0; j < subRight; j++){
		rightArray[j] = array[mid + 1 + j];
	}

	int indexOfSubLeft = 0; // Initial index of left sub-array
	int	indexOfSubRight = 0; // Initial index of right sub-array
	int indexOfMergedArray = left; // Initial index of merged array

	// Merge the temp arrays back into array[left..right]
	while (indexOfSubLeft < subLeft && indexOfSubRight < subRight && typeOfSort ==
1) {
		if (leftArray[indexOfSubLeft].bookTitle <=
rightArray[indexOfSubRight].bookTitle) {
			array[indexOfMergedArray] = leftArray[indexOfSubLeft];
			indexOfSubLeft++;
		}
		else {
			array[indexOfMergedArray] = rightArray[indexOfSubRight];
			indexOfSubRight++;
```

```
				}
				indexOfMergedArray++;
		}

		// Merge the temp arrays back into array[left..right]
		while (indexOfSubLeft < subLeft && indexOfSubRight < subRight && typeOfSort ==
2) {
				if (leftArray[indexOfSubLeft].author <= rightArray[indexOfSubRight].author) {
						array[indexOfMergedArray] = leftArray[indexOfSubLeft];
						indexOfSubLeft++;
				}
				else {
						array[indexOfMergedArray] = rightArray[indexOfSubRight];
						indexOfSubRight++;
				}
				indexOfMergedArray++;
		}

		// Merge the temp arrays back into array[left..right]
		while (indexOfSubLeft < subLeft && indexOfSubRight < subRight && typeOfSort ==
3) {
				if (leftArray[indexOfSubLeft].yearOfPublish >=
rightArray[indexOfSubRight].yearOfPublish) {
						array[indexOfMergedArray] = leftArray[indexOfSubLeft];
						indexOfSubLeft++;
				}
				else {
						array[indexOfMergedArray] = rightArray[indexOfSubRight];
						indexOfSubRight++;
				}
				indexOfMergedArray++;
		}

		// Copy the remaining elements of
		// left[], if there are any
		while (indexOfSubLeft < subLeft) {
				array[indexOfMergedArray] = leftArray[indexOfSubLeft];
				indexOfSubLeft++;
				indexOfMergedArray++;
		}

		// Copy the remaining elements of
		// right[], if there are any
		while (indexOfSubRight < subRight) {
				array[indexOfMergedArray] = rightArray[indexOfSubRight];
				indexOfSubRight++;
				indexOfMergedArray++;
```

```
        }
}
```

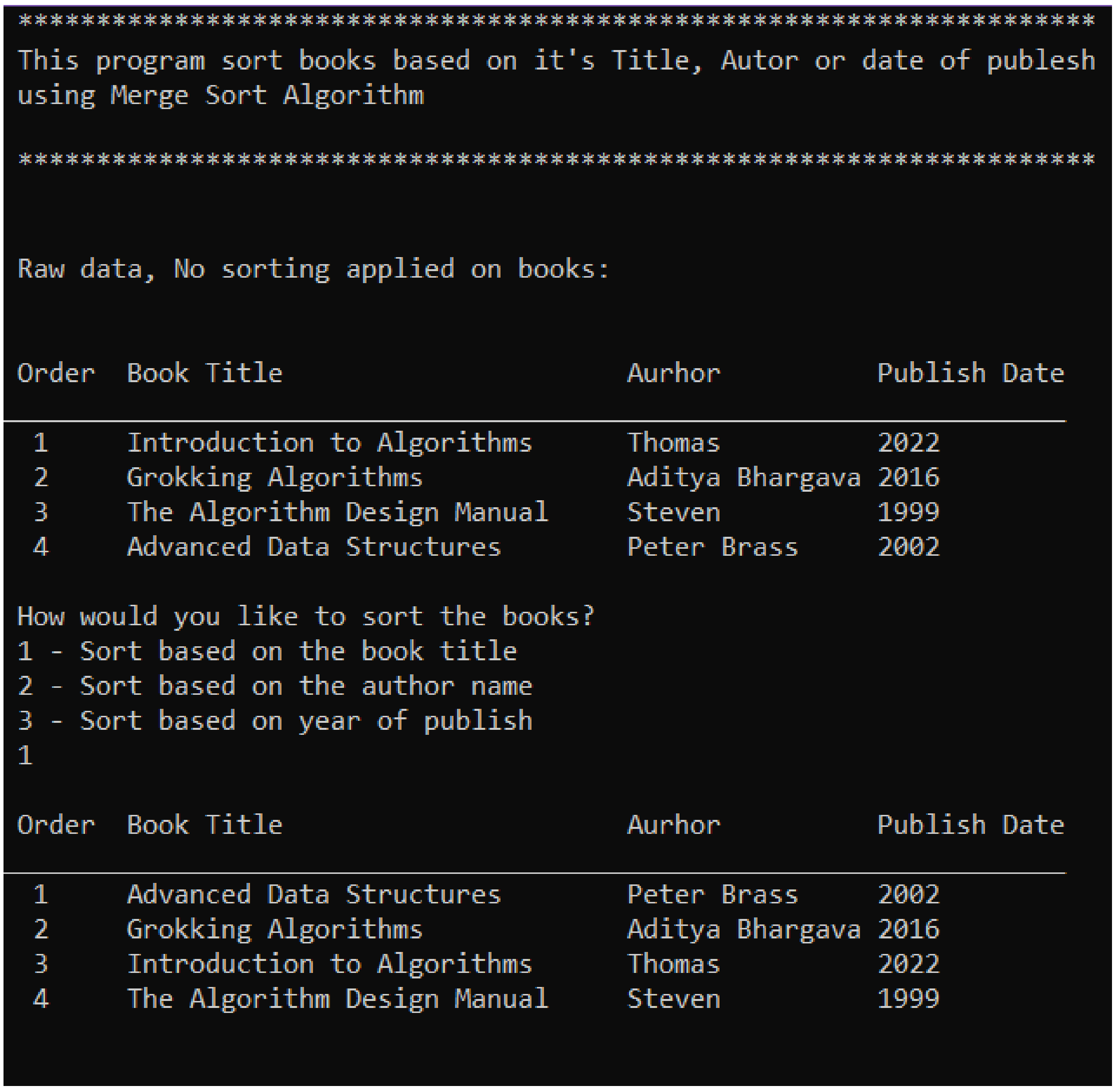
Input/Output Result

The program initialize all books during the execution time,



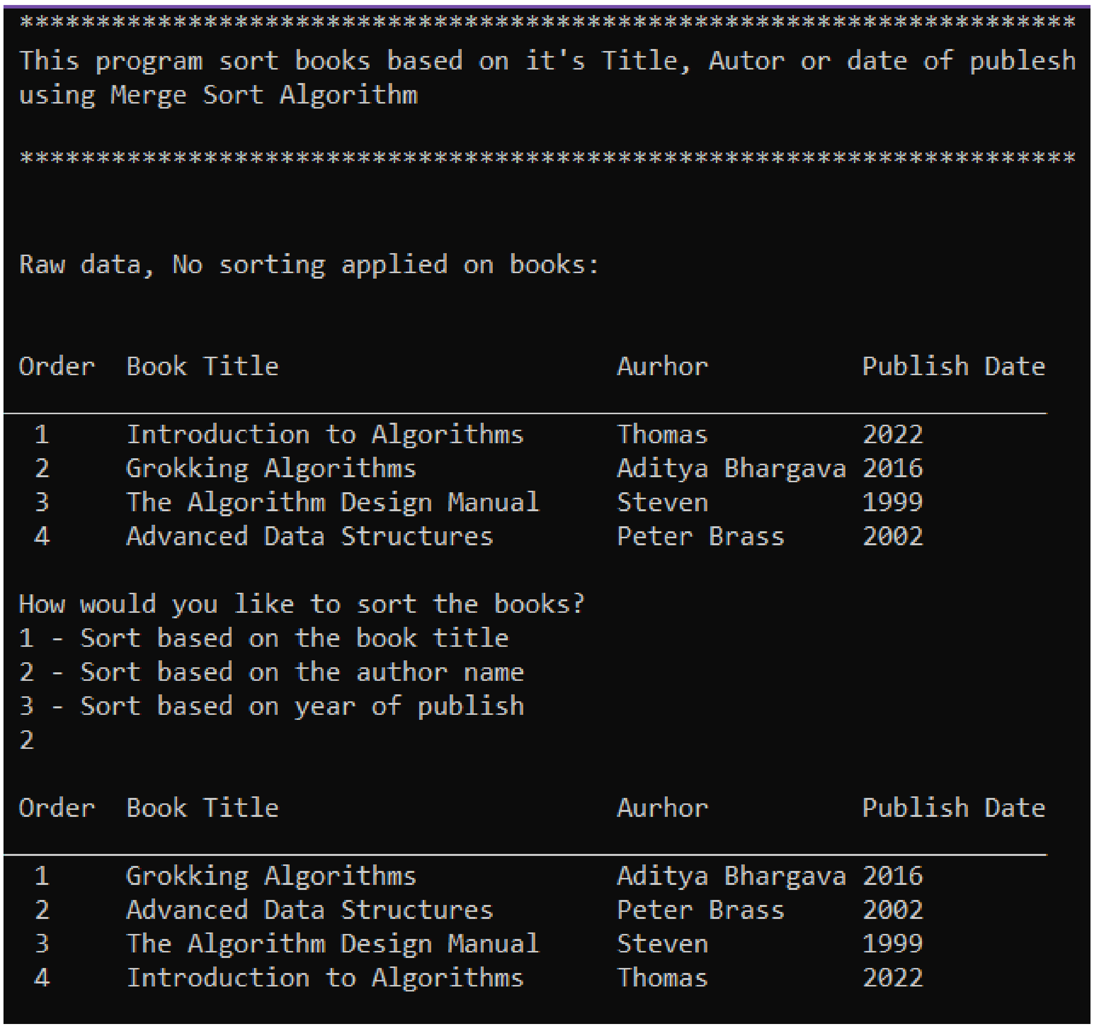
```
//initilize the books of array
int n = 4;
Book array[n];

// Details of Book 1
array[0].bookTitle = "Introduction to Algorithms";
array[0].author = "Thomas    ";
array[0].yearOfPublish = 2022;

// Details of Book 2
array[1].bookTitle = "Grokking Algorithms    ";
array[1].author = "Aditya Bhargava";
array[1].yearOfPublish = 2016;

// Details of Book 3
array[2].bookTitle = "The Algorithm Design Manual";
array[2].author = "Steven    ";
array[2].yearOfPublish = 1999;

// Details of Book 4
array[3].bookTitle = "Advanced Data Structures";
array[3].author = "Peter Brass";
array[3].yearOfPublish = 2002;
```

The program reads the input from the user to know which type of sorting would like to perform.
If the user chose 1 the book will be sorted based on Book’s Title as shown below:



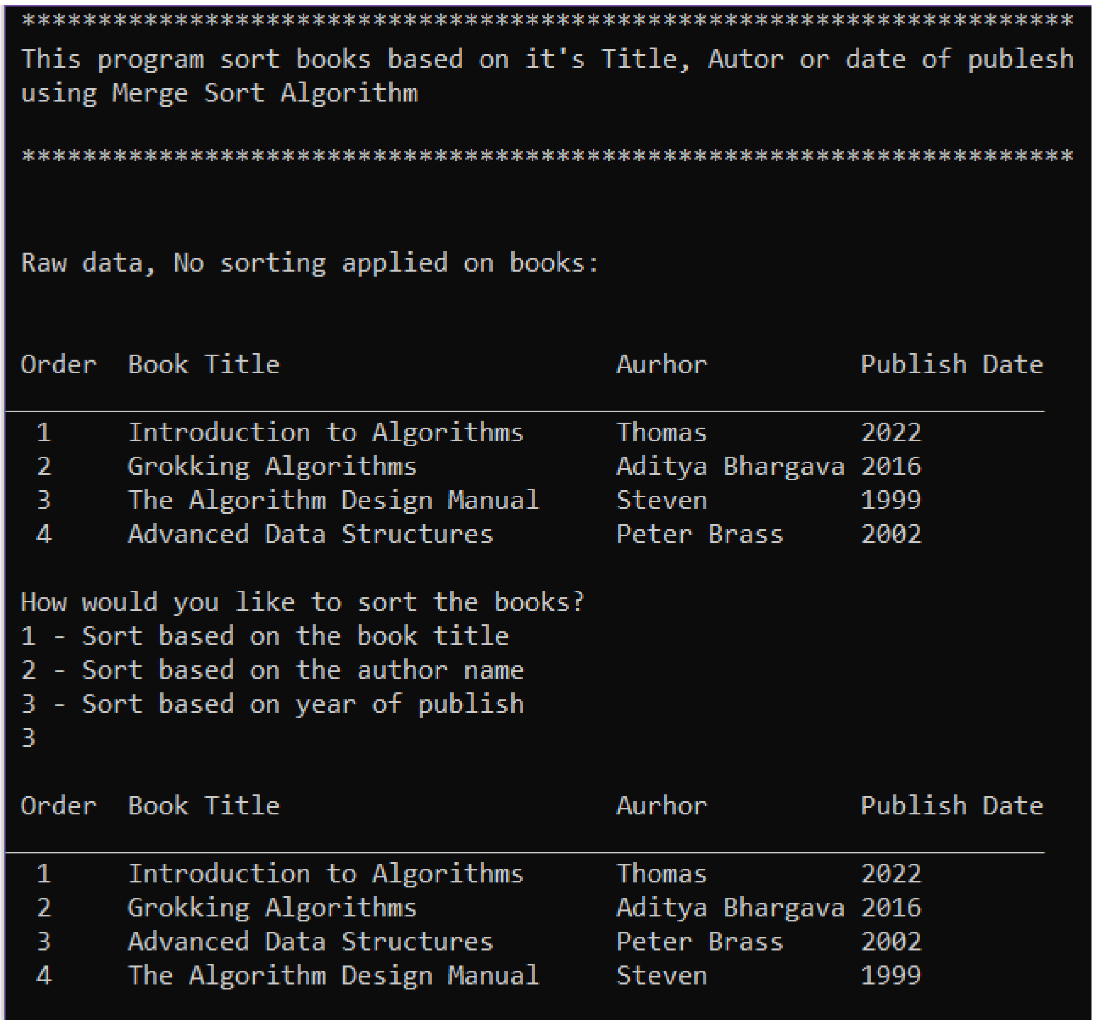
```
**********************************************************************
This program sort books based on it's Title, Autor or date of publesh
using Merge Sort Algorithm

**********************************************************************


Raw data, No sorting applied on books:


Order  Book Title                      Aurhor          Publish Date
----------------------------------------------------------------------
 1     Introduction to Algorithms      Thomas          2022
 2     Grokking Algorithms             Aditya Bhargava 2016
 3     The Algorithm Design Manual     Steven          1999
 4     Advanced Data Structures        Peter Brass     2002

How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
1

Order  Book Title                      Aurhor          Publish Date
----------------------------------------------------------------------
 1     Advanced Data Structures        Peter Brass     2002
 2     Grokking Algorithms             Aditya Bhargava 2016
 3     Introduction to Algorithms      Thomas          2022
 4     The Algorithm Design Manual     Steven          1999
```

If the user chose 2 the book will be sorted based on Book’s Author as shown below:



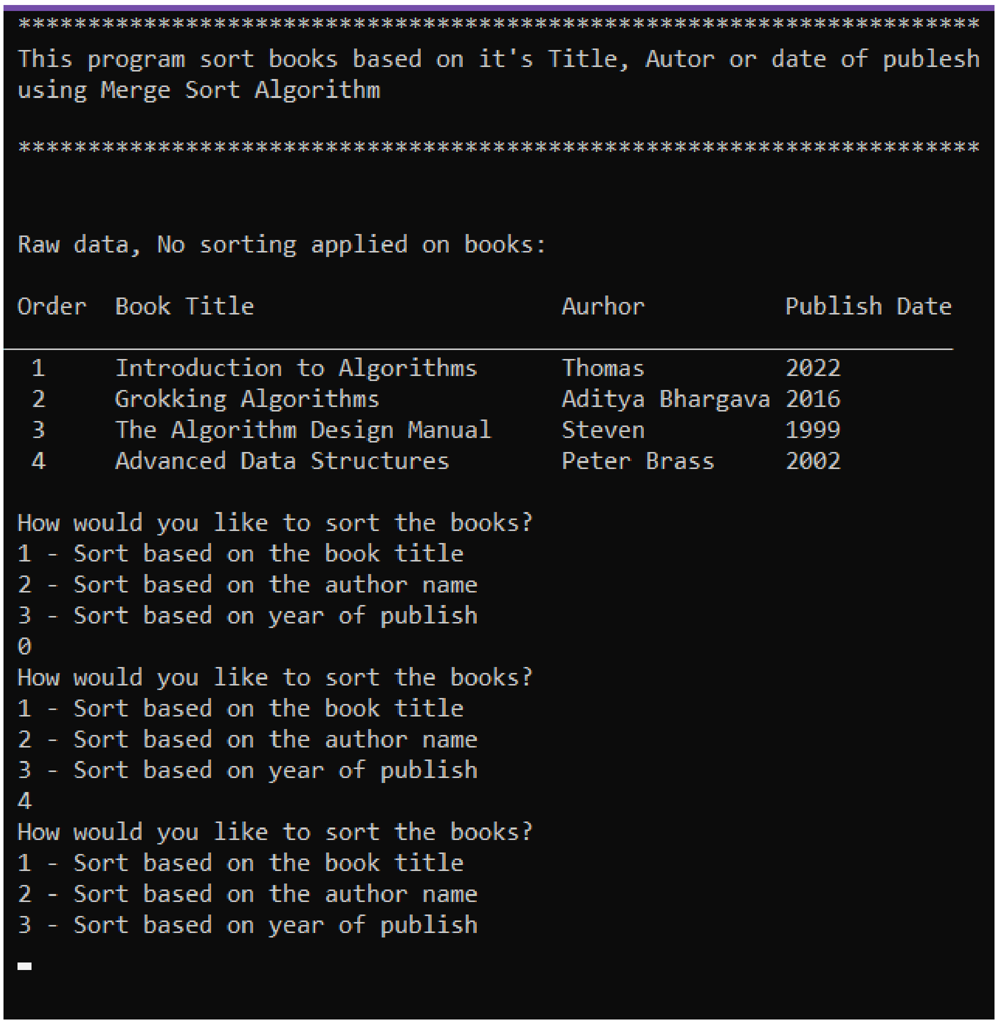
```
*********************************************************************
This program sort books based on it's Title, Autor or date of publesh
using Merge Sort Algorithm

*********************************************************************



Raw data, No sorting applied on books:


Order  Book Title                      Aurhor           Publish Date

  1      Introduction to Algorithms      Thomas           2022
  2      Grokking Algorithms             Aditya Bhargava  2016
  3      The Algorithm Design Manual     Steven           1999
  4      Advanced Data Structures        Peter Brass      2002

How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
2

Order  Book Title                      Aurhor           Publish Date

  1      Grokking Algorithms             Aditya Bhargava  2016
  2      Advanced Data Structures        Peter Brass      2002
  3      The Algorithm Design Manual     Steven           1999
  4      Introduction to Algorithms      Thomas           2022
```

If the user chooses 3 the book will be sorted based on the Book’s Year of Publish as shown below:



```
**********************************************************************
This program sort books based on it's Title, Autor or date of publesh
using Merge Sort Algorithm

**********************************************************************



Raw data, No sorting applied on books:


Order  Book Title                      Aurhor          Publish Date
---------------------------------------------------------------------
 1     Introduction to Algorithms      Thomas          2022
 2     Grokking Algorithms             Aditya Bhargava 2016
 3     The Algorithm Design Manual     Steven          1999
 4     Advanced Data Structures        Peter Brass     2002

How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
3

Order  Book Title                      Aurhor          Publish Date
---------------------------------------------------------------------
 1     Introduction to Algorithms      Thomas          2022
 2     Grokking Algorithms             Aditya Bhargava 2016
 3     Advanced Data Structures        Peter Brass     2002
 4     The Algorithm Design Manual     Steven          1999
```

If the user chooses any numbers other than 1,2,3 the program will prompt the user to insert again as shown below.



```
*********************************************************************
This program sort books based on it's Title, Autor or date of publesh
using Merge Sort Algorithm

*********************************************************************


Raw data, No sorting applied on books:

Order  Book Title                      Aurhor          Publish Date
--------------------------------------------------------------------
 1     Introduction to Algorithms      Thomas          2022
 2     Grokking Algorithms             Aditya Bhargava 2016
 3     The Algorithm Design Manual     Steven          1999
 4     Advanced Data Structures        Peter Brass     2002

How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
0
How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
4
How would you like to sort the books?
1 - Sort based on the book title
2 - Sort based on the author name
3 - Sort based on year of publish
```

Conclusion

Libraries serve an important role in society as portals to information and culture. The tools and services they provide foster learning, encourage literacy and education, and aid in the formation of new ideas and views that are essential to a creative and dynamic society. Students may access online books, photos, videos, and any other instructional items by using a digital library instead of having to wait and go to the nearest physical library. They can do it in a structured setting, such as school, or they can relax at home and gain fast access to the knowledge they want. Furthermore, using our system, students may locate the category of the book they desire based on its Title, Author, or Date of Publication, and library management can quickly rearrange the books by following what our system sorted.

References

- Digital Libraries: Challenges and Problems by dr Namita Khot