THE UNIVERSITY OF LAHORE

Submitted by: Muhammad Raffey

Submitted to: Sir Nouman

Sap ID: 70153209

Department: CS

Section: 4G

Assignment no: 1

The University of Lahore

Department of Computer Science & IT

CS-09204 Data Structures and Algorithm
Fall 2025

Assignment # 1a

Participant ID #	70153209	**CLO: 2** **PLO:**
Total Marks:	40	**Obtained Marks:**

Instructions:

Analyze the following C++ code snippets. For each code snippet, calculate its time complexity. Explain your reasoning and, if needed, describe how the time complexity is derived. Write down the time complexity in Big O notation (e.g., O(1), O(n), O(log n), O(n^2)).

Code Snippet 1

C++ Code	**Time Complexity Analysis**

int main() {
int a = 10;
int b = 20;
int result = a + b; // Single operation
cout << "Result: " << result << endl;
return 0;
}
	O(1)
	single line

Code Snippet 2

C++ Code	**Time Complexity Analysis**

int main() {
int n = 100;
int sum = 0;
for (int i = 1; i <= n; i++) { // Loop from 1 to n
sum += i;
}
cout << "Sum: " << sum << endl;
return 0;}
	O(n)
	1 → n loop

Code Snippet 3

C++ Code	Time Complexity Analysis

int main() {
int n = 5;
for (int i = 0; i < n; i++) { // Outer loop
for (int j = 0; j < n; j++) { // Inner loop
cout << i * j << endl;
}
}
return 0;
}
	$O(n^2)$
	Nested loop

Code Snippet 4

C++ Code	Time Complexity Analysis

int binarySearch(int arr[], int n, int target) {
int low = 0, high = n - 1;
while (low <= high) {
int mid = low + (high - low) / 2;
if (arr[mid] == target) return mid; // Element found
else if (arr[mid] < target) low = mid + 1;
else high = mid - 1;
}
return -1; // Element not found
}
	$O(\log n)$
	fractional
	Increment

Code Snippet 5

$O(\log n)$

C++ Code	Time Complexity Analysis

int main() {
int n = 10;
for (int i = 1; i <= n; i *= 2) { // Logarithmic loop
for (int j = 1; j <= n; j++) { // Linear loop inside
cout << i + j << endl;
$O(n)$
}
}
return 0;}
	$O(n \log n)$

<u>Part 1</u>

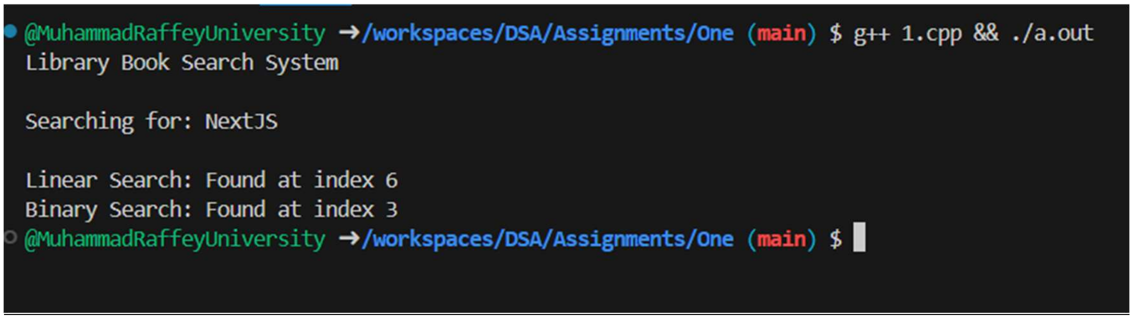
<u>Task1:</u> Implement a Library Book Search System using Arrays

<u>Code:</u>

```
#include <iostream>
#include <string>
using namespace std;

bool LinearSearch(string arr[], int n, string target, int &index)
{
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == target)
        {
            index = i;
            return true;
        }
    }
    return false;
}

bool binarySearch(string arr[], int n, string target, int &index)
{
    int low = 0;
    int high = n - 1;
    while (low <= high)
    {
        int mid = (low + high) / 2;
        if (arr[mid] == target)
        {
            index = mid;
            return true;
        }
        else if (arr[mid] < target)
        {
            low = mid + 1;
        }
        else
        {
            high = mid - 1;
        }
    }
    return false;
}

void bubbleSort(string arr[], int n)
{
    bool sorted;
    int i = 0;
    do
    {
        sorted = false;
        for (int j = 0; j < n - 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
                sorted = true;
            }
        }
        i++;
    } while (sorted);
}

int main()
```

```cpp
{
    string books[] = {"DSA", "Linear", "Theory of Automata", "C++",
                      "Python", "TypeScript", "NextJS"};
    int n = sizeof(books) / sizeof(books[0]);

    string sortedBooks[8];
    for (int i = 0; i < n; i++)
    {
        sortedBooks[i] = books[i];
    }
    bubbleSort(sortedBooks, n);

    string searchBook = "NextJS";
    int index;

    cout << "Library Book Search System\n\n";
    cout << "Searching for: " << searchBook << "\n\n";

    if (LinearSearch(books, n, searchBook, index))
    {
        cout << "Linear Search: Found at index " << index << "\n";
    }
    else
    {
        cout << "Linear Search: Not found\n";
    }

    if (binarySearch(sortedBooks, n, searchBook, index))
    {
        cout << "Binary Search: Found at index " << index << "\n";
    }
    else
    {
        cout << "Binary Search: Not found\n";
    }

    return 0;
}
```

Output:



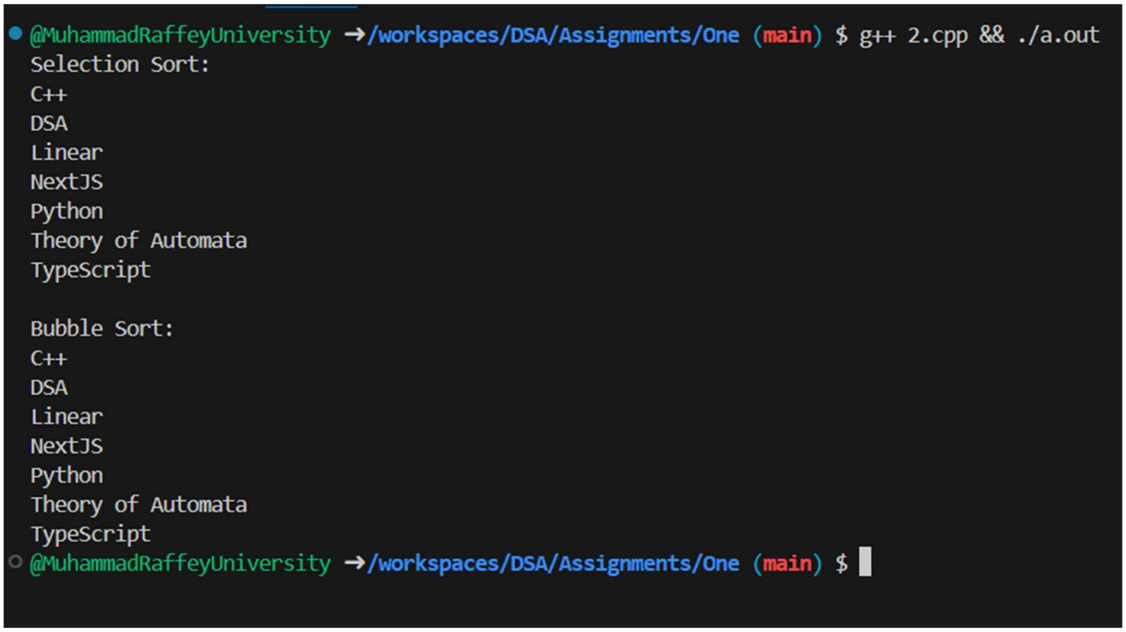
```
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $ g++ 1.cpp && ./a.out
Library Book Search System

Searching for: NextJS

Linear Search: Found at index 6
Binary Search: Found at index 3
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $
```

Task 2: Sort the Library

Code:

```
#include <iostream>
#include <string>
using namespace std;

void selectionSort(string arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        int minIdx = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[minIdx])
            {
                minIdx = j;
            }
        }
        if (minIdx != i)
        {
            swap(arr[i], arr[minIdx]);
        }
    }
}

void bubbleSort(string arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
}

void display(string arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << endl;
    }
}

int main()
{
    string books[] = {"DSA", "Linear", "Theory of Automata", "C++",
                        "Python", "TypeScript", "NextJS"};

    int n = sizeof(books) / sizeof(books[0]);

    cout << "Selection Sort:\n";
    selectionSort(books, n);
    display(books, n);

    cout << "\nBubble Sort:\n";
    bubbleSort(books, n);
    display(books, n);

    return 0;
}
```

<u>Output:</u>



```
● @MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $ g++ 2.cpp && ./a.out
  Selection Sort:
  C++
  DSA
  Linear
  NextJS
  Python
  Theory of Automata
  TypeScript

  Bubble Sort:
  C++
  DSA
  Linear
  NextJS
  Python
  Theory of Automata
  TypeScript
○ @MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $ 
```

Part 2

Task3: **Implement a Library Book Search System using a link list**

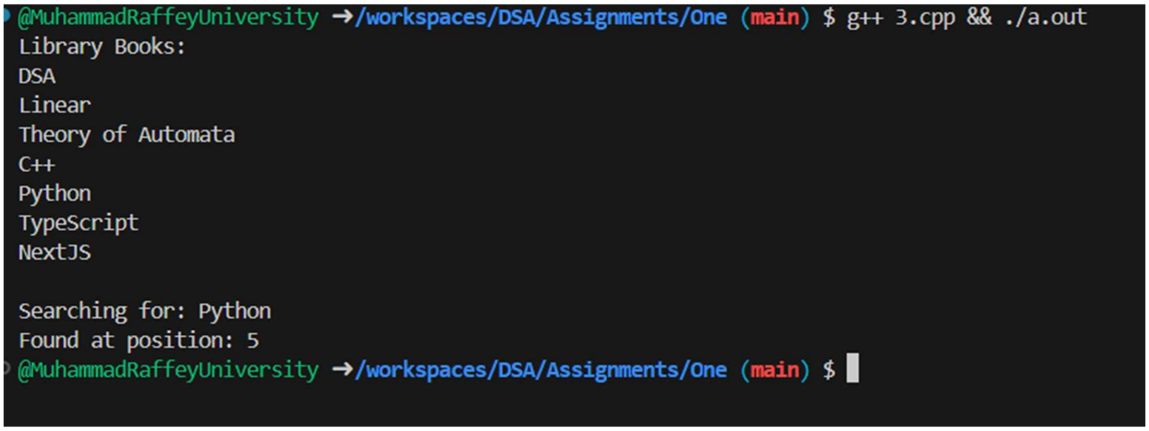
Code:

```
#include <iostream>
#include <string>
using namespace std;

class Node
{
public:
    string data;
    Node *next;
    Node(string val) : data(val), next(NULL) {}
};

class LinkedList
{
private:
    Node *head;

public:
    LinkedList()
    {
        head = NULL;
    }

    void insertAtEnd(string value)
    {
        Node *newNode = new Node(value);

        if (head == NULL)
        {
```

```
        head = newNode;
        return;
    }

    Node *temp = head;
    while (temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = newNode;
}

int searchBook(string bookTitle)
{
    Node *temp = head;
    int position = 1;

    while (temp != NULL)
    {
        if (temp->data == bookTitle)
        {
            return position;
        }
        temp = temp->next;
        position++;
    }
    return -1;
}

void display()
{
    Node *temp = head;
```

```
        while (temp != NULL)
        {
            cout << temp->data << endl;
            temp = temp->next;
        }
    }
};

int main()
{
    LinkedList library;

    library.insertAtEnd("DSA");
    library.insertAtEnd("Linear");
    library.insertAtEnd("Theory of Automata");
    library.insertAtEnd("C++");
    library.insertAtEnd("Python");
    library.insertAtEnd("TypeScript");
    library.insertAtEnd("NextJS");

    cout << "Library Books:\n";
    library.display();

    string searchBook = "Python";
    int position = library.searchBook(searchBook);

    cout << "\nSearching for: " << searchBook << endl;
    if (position != -1)
    {
        cout << "Found at position: " << position << endl;
    }
    else
    {
        cout << "Not found" << endl;
    }

    return 0;
}
```

Output:



```
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $ g++ 3.cpp && ./a.out
Library Books:
DSA
Linear
Theory of Automata
C++
Python
TypeScript
NextJS

Searching for: Python
Found at position: 5
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $
```

Task4 Library Management System

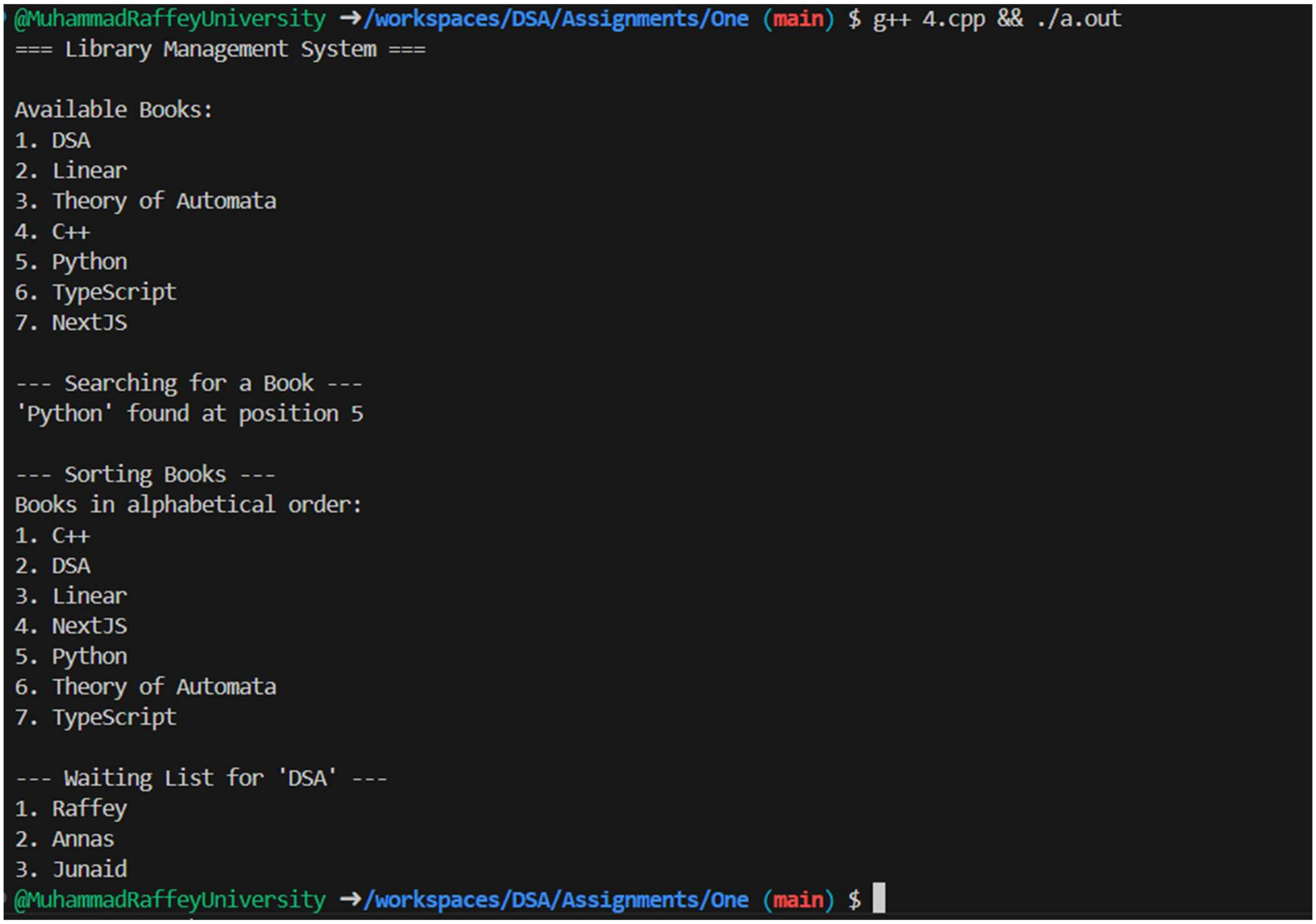
Code:

```
#include <iostream>
#include <string>
using namespace std;

class Node
{
public:
    string data;
    Node *next;
    Node(string val) : data(val), next(NULL) {}
};

class WaitingList
{
```

```
private:
    Node *head;

public:
    WaitingList()
    {
        head = NULL;
    }

    void addToWaitingList(string name)
    {
        Node *newNode = new Node(name);
        if (head == NULL)
        {
            head = newNode;
            return;
        }
        Node *temp = head;
        while (temp->next != NULL)
        {
            temp = temp->next;
        }
        temp->next = newNode;
    }

    void displayWaitingList()
    {
        if (head == NULL)
        {
            cout << "No one in waiting list\n";
            return;
        }
        Node *temp = head;
        int position = 1;
        while (temp != NULL)
        {
            cout << position << ". " << temp->data << endl;
            temp = temp->next;
            position++;
        }
    }
};

bool searchBook(string arr[], int n, string target, int &index)
{
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == target)
        {
            index = i;
            return true;
        }
    }
    return false;
}

void bubbleSort(string arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - 1 - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
}

void printBooks(string arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << i + 1 << ". " << arr[i] << endl;
    }
}

int main()
{
    string books[] = {"DSA", "Linear", "Theory of Automata", "C++",
                      "Python", "TypeScript", "NextJS"};
    int n = 7;
    WaitingList waitingList;
    cout << "=== Library Management System ===\n\n";
    cout << "Available Books:\n";
    printBooks(books, n);
    cout << "\n--- Searching for a Book ---\n";
    string searchTitle = "Python";
    int index;
    if (searchBook(books, n, searchTitle, index))
    {cout << "'" << searchTitle << "' found at position " << (index + 1) << endl;}
    else
    {cout << "Book not found\n";}
    cout << "\n--- Sorting Books ---\n";
    string sortedBooks[7];
```

```
for (int i = 0; i < n; i++)

{sortedBooks[i] = books[i];}

bubbleSort(sortedBooks, n);

cout << "Books in alphabetical order:\n";

printBooks(sortedBooks, n);

cout << "\n--- Waiting List for 'DSA' ---\n";

waitingList.addToWaitingList("Raffey");

waitingList.addToWaitingList("Annas");

waitingList.addToWaitingList("Junaid");

waitingList.displayWaitingList();

return 0;}
```

Output:



```
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $ g++ 4.cpp && ./a.out
=== Library Management System ===

Available Books:
1. DSA
2. Linear
3. Theory of Automata
4. C++
5. Python
6. TypeScript
7. NextJS

--- Searching for a Book ---
'Python' found at position 5

--- Sorting Books ---
Books in alphabetical order:
1. C++
2. DSA
3. Linear
4. NextJS
5. Python
6. Theory of Automata
7. TypeScript

--- Waiting List for 'DSA' ---
1. Raffey
2. Annas
3. Junaid
@MuhammadRaffeyUniversity →/workspaces/DSA/Assignments/One (main) $
```