Name: Ankush Singh
Group: 5C13
Roll No.: 040

Date: 19-09-2023

EXPERIMENT- 2

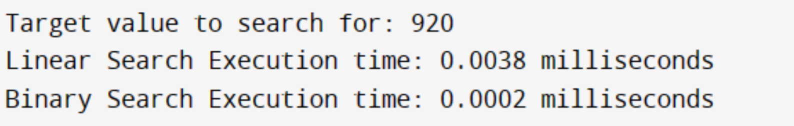
Aim:

To implement Linear search and Binary search and analyse its time complexity.

Code:

```
#include <bits/stdc++.h>
#include <ctime>

using namespace std;

int linearSearch(int arr[], int n, int target) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == target) {
            return i;
        }
    }
    return -1;
}

int binarySearch(int arr[], int n, int target) {
    int left = 0;
    int right = n - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] == target) {
            return mid;
        } else if (arr[mid] < target) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    return -1;
}

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

int main() {
    const int arraySize = 1000;
    int arr[arraySize];

    // Fill the array with numbers from 1 to 1000
    for (int i = 0; i < arraySize; i++) {
        arr[i] = i + 1;
    }

    int target = 920;
    int numTrials = 5;
    cout << "Target value to search for: " << target << endl << endl;

    double totalLinearExecutionTime = 0.0;
    double totalBinaryExecutionTime = 0.0;

    for (int trial = 0; trial < numTrials; ++trial) {
        // Linear Search
        clock_t start_time = clock();
        int linearResult = linearSearch(arr, arraySize, target);
        clock_t end_time = clock();

        double linearExecutionTime = (end_time - start_time) * 1000.0 / CLOCKS_PER_SEC;
        totalLinearExecutionTime += linearExecutionTime;

        // Binary Search
        start_time = clock();
        int binaryResult = binarySearch(arr, arraySize, target);
        end_time = clock();

        double binaryExecutionTime = (end_time - start_time) * 1000.0 / CLOCKS_PER_SEC;
        totalBinaryExecutionTime += binaryExecutionTime;
    }

    double avgLinearExecutionTime = totalLinearExecutionTime / numTrials;
    cout << "Linear Search Execution time: " << avgLinearExecutionTime << " milliseconds"
<< endl;

    double avgBinaryExecutionTime = totalBinaryExecutionTime / numTrials;
    cout << "Binary Search Execution time: " << avgBinaryExecutionTime << " milliseconds"
<< endl;

    return 0;
}
```

Output:



```
Target value to search for: 920
Linear Search Execution time: 0.0038 milliseconds
Binary Search Execution time: 0.0002 milliseconds
```