Lab Sheet 4: Compare Searching Algorithms – Linear Search and Binary Search

Aim:

To implement Linear Search and Binary Search algorithms in C and to compare their performance for searching an

element in an array.

Procedure:

- Open Turbo C and create a new C program.
- Include the required header file stdio.h.
- Read the number of elements and array values from the user.
- Read the search key to be found.
- Implement the Linear Search algorithm to search for the key in the array.
- Display whether the element is found and its position.
- Sort the array before applying Binary Search.
- Implement the Binary Search algorithm on the sorted array.
- Display the search result obtained using Binary Search.
- Compile and run the program.
- Observe and compare the results of both searching techniques.

Program: Linear Search and Binary Search

```
#include <stdio.h>
#include <time.h>
// Linear Search Function
int linearSearch(int a[], int n, int key) {
for (int i = 0; i < n; i++) {
if (a[i] == key)
return i;
}
return -1;
}
// Binary Search Function
```

```
int binarySearch(int a[], int n, int key) {
int low = 0, high = n - 1, mid;
while (low <= high) {
mid = (low + high) / 2;
if (a[mid] == key)
return mid;
else if (a[mid] < key)
low = mid + 1;
else
high = mid - 1;
}
return -1;
}
// Function to sort array (for binary search)
void sortArray(int a[], int n) {
int temp;
for (int i = 0; i < n - 1; i++) {
for (int j = i + 1; j < n; j++) {
if (a[i] > a[j]) {
temp = a[i];
a[i] = a[j];
a[j] = temp;
}
}
}
}
int main() {
int a[20], n, key, pos;
clock_t start, end;
```

```
double time_taken;
// Input array
printf("Enter number of elements: ");
scanf("%d", &n);
printf("Enter array elements:\n");
for (int i = 0; i < n; i++)
scanf("%d", &a[i]);
printf("Enter element to search: ");
scanf("%d", &key);
// Linear Search
start = clock();
pos = linearSearch(a, n, key);
end = clock();
time_taken = ((double)(end - start)) / CLOCKS_PER_SEC;
if (pos != -1)
printf("Linear Search: Element found at position %d\n", pos + 1);
else
printf("Linear Search: Element not found\n");
printf("Time taken by Linear Search: %f seconds\n", time_taken);
// Sort array for Binary Search
sortArray(a, n);
// Binary Search
start = clock();
pos = binarySearch(a, n, key);
end = clock();
time_taken = ((double)(end - start)) / CLOCKS_PER_SEC;
if (pos != -1)
printf("Binary Search: Element found at position %d\n", pos + 1);
else
```

```
printf("Binary Search: Element not found\n");
printf("Time taken by Binary Search: %f seconds\n", time_taken);
return 0;
}
```

Result:

Linear Search and Binary Search algorithms were implemented successfully. Linear Search works on both sorted and

unsorted arrays but takes more time for large inputs, whereas Binary Search is faster with O(log n) time complexity

but requires the array to be sorted