|  |  |
| --- | --- |
| https://lh6.googleusercontent.com/NfxikauFutj0NGA6tZLo-9OYuuo3IKL8zEJJ1vcz211S75EzE4cV8OiG_9xiNDI7UXEpwJU3L0HucAlThGrGrN0dBZNqE1lpoGJHT3oGyRmOalINq0YRgVxjjMgLbZ3n8Pq0wpa0thnjQcjh_kKTHUlnU-ExSfeM05HoXY1-s1p-Qtg0OyqjFjrwB05Nj4mxriktv4A | SVKM’s NMIMS  School of Technology Management & Engineering Navi Mumbai Campus |
| Department of Computer Engineering |

|  |  |
| --- | --- |
| Name: Yash Patil | SAP ID: 70472200140 |
| Semester: III | Year: II |
| Subject: Data Structures and Algorithm | Roll No.: A176 |
| Practical: 10 | Batch: 1 |

**Aim:–**

Implementation of Linear search and binary search, understand their difference

**Theory:–**

Linear Search and Binary Search are two fundamental searching algorithms used to find an element in a collection (usually an array) of data. They differ significantly in terms of their approach and efficiency:

Linear Search:

- Algorithm: Linear search, also known as sequential search, examines each element in the data structure one by one until the target element is found or all elements are exhausted.

- Efficiency: It has a time complexity of O(n), where 'n' is the number of elements in the collection. Linear search has a linear relationship with the number of elements. It is suitable for small collections or when the elements are not sorted.

- Advantages: Simple to understand and easy to implement. It works for both sorted and unsorted data.

- Disadvantages: Inefficient for large collections as it checks each element in turn. It is not optimized for sorted data.

Binary Search:

- Algorithm: Binary search is applicable only to sorted collections. It starts by comparing the target element with the middle element of the collection. If they match, the search is successful. If the target is less than the middle element, it narrows the search to the lower half of the collection; if the target is greater, it narrows it to the upper half. This process is repeated on the selected half until the element is found or it is determined that the element is not in the collection.

- Efficiency: Binary search has a time complexity of O(log n), making it significantly more efficient than linear search for large collections. It has a logarithmic relationship with the number of elements.

- Advantages: Highly efficient for sorted collections. It quickly reduces the search space to find the target element.

- Disadvantages: It requires the collection to be sorted. If the collection is frequently modified, the sorting step can be expensive.

Key Differences:

1. Data Order: Linear search can be used on both sorted and unsorted data, while binary search requires sorted data.

2. Time Complexity: Linear search has a time complexity of O(n), while binary search has a time complexity of O(log n), making binary search significantly faster for large datasets.

3. Efficiency: Binary search is an efficient search algorithm for large datasets because it quickly narrows down the search space.

4. Implementation: Linear search is straightforward to implement, while binary search is more complex and requires the data to be sorted.

**Code/Implementation –**

#include <stdio.h>

int linearSearch(int arr[], int n, int key) {

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

if (arr[i] == key) {

return i;

}

}

return -1;

}

int binarySearch(int arr[], int n, int key) {

int left = 0;

int right = n - 1;

while (left <= right) {

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

if (arr[mid] == key) {

return mid;

}

if (arr[mid] < key) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

int main() {

int n, choice, key, index;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

scanf("%d", &arr[i]);

}

do {

printf("\nMenu:\n");

printf("1. Linear Search\n");

printf("2. Binary Search\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter the key to search: ");

scanf("%d", &key);

index = linearSearch(arr, n, key);

if (index != -1) {

printf("Key found at index %d\n", index);

} else {

printf("Key not found in the array\n");

}

break;

case 2:

printf("Enter the key to search: ");

scanf("%d", &key);

index = binarySearch(arr, n, key);

if (index != -1) {

printf("Key found at index %d\n", index);

} else {

printf("Key not found in the array\n");

}

break;

case 3:

printf("Exiting the program.\n");

break;

default:

printf("Invalid choice. Please enter a valid option.\n");

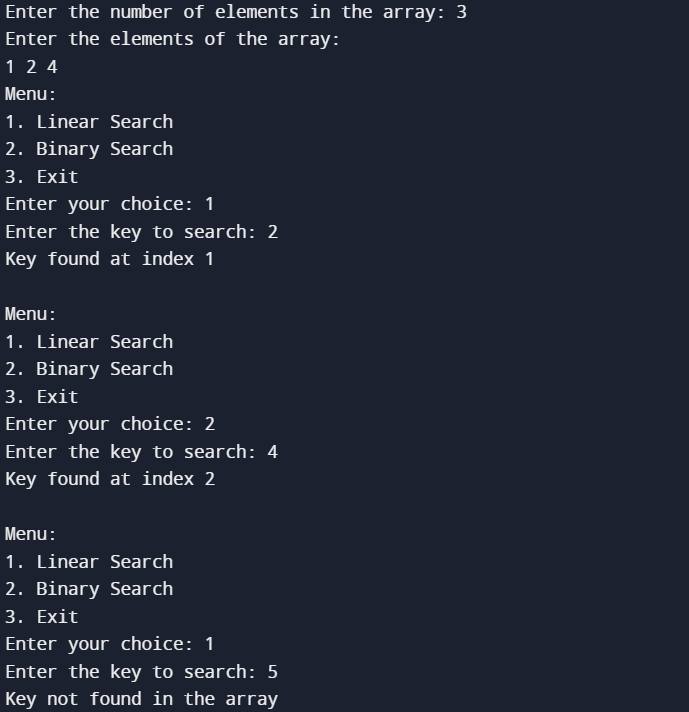
}

} while (choice != 3);

return 0;

}

**Output:-**



**Conclusion:-**

*In conclusion, Linear Search is a straightforward and versatile search algorithm suitable for both sorted and unsorted data, but its efficiency diminishes with large datasets. On the other hand, Binary Search is highly efficient for large datasets, but it requires the data to be sorted. The choice between the two depends on the nature of the data and the trade-off between simplicity and efficiency.*