

Searching Algorithms

Searching is an algorithm used to find a specific data element within a dataset. In this module, we will discuss two common searching methods: Linear Search and Binary Search.

Linear Search

The simplest and most commonly used searching algorithm is the sequential or linear search. The way this algorithm works is by comparing the search key with each element in the list sequentially until the desired data is found or until all elements have been checked.

Key	List
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8
3	6 4 1 9 7 3 2 8

Below is an illustration of the linear search process. The search key is the number 3, and it is compared one by one until the number 3 is found.

Example Program using Linear Search:

```
#include <stdio.h>

#define SIZE 10

int main() {
    int arr[SIZE] = {15, 7, 2, 10, 27, 77, 32, 43, 69, 56};
    int i, input;
```

```

printf("Enter the number to search: ");
scanf("%d", &input);

for (i = 0; i < SIZE; i++) {
    if (arr[i] == input) {
        printf("\n%d is located at position %d\n", input, i+1);
        break;
    }
}

if (i == SIZE) {
    printf("\nNot found in the list!\n");
}

return 0;
}

```

Binary Search

Binary search is an algorithm that searches for a data element within a sorted list. This method is faster than Linear Search but requires the list to be sorted in ascending order beforehand. The algorithm repeatedly divides the search range into two halves. If the search key is smaller than the middle element, the search continues in the left half. If the search key is larger, the search continues in the right half.



Below is an illustration of the binary search process. The input key is 8, so the search range is reduced to the right half of the list.

Example Program using Binary Search:

```
#include <stdio.h>

#define SIZE 7

int main() {
    int arr[SIZE] = {3, 8, 16, 29, 32, 47, 66};
    int left, right, middle, input;

    printf("Enter the number to search: ");
    scanf("%d", &input);

    left = 0;
    right = SIZE - 1;

    while (left <= right) {
        middle = (left + right) / 2;

        // Check if the middle element is the target
        if (arr[middle] == input) {
            printf("\n%d is located at position %d\n", input, middle+1);
            break;
        }

        // Adjust search range
        if (arr[middle] < input)
            left = middle + 1;
        else
            right = middle - 1;
    }

    if (left > right)
        printf("\nNot found in the list!\n");

    return 0;
}
```

Revision #1

Created 21 February 2025 23:50:42 by YP

Updated 21 February 2025 23:58:32 by YP