

Searching Algorithm

Teknik Informatika UMJ

Dosen Pengampu: Mirza Sutrisno, M.Kom.



UMJ
UNIVERSITAS
MUHAMMADIYAH
JAKARTA

Enlightening, Empowering

Linear Search

Linear search adalah algoritma pencarian berurutan di mana kita mulai dari satu ujung dan memeriksa setiap elemen daftar hingga elemen yang diinginkan ditemukan. Ini adalah algoritma pencarian paling sederhana.

Cara Kerja

Bagaimana Pencarian Linier Bekerja?

Langkah-langkah berikut diikuti untuk mencari elemen $k = 1$ dalam daftar di bawah ini.

2	4	0	1	9
---	---	---	---	---

Array yang akan dicari

1. Mulai dari elemen pertama, bandingkan k dengan setiap elemen X .

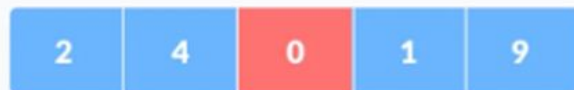
$k = 1$



$k \neq 2$



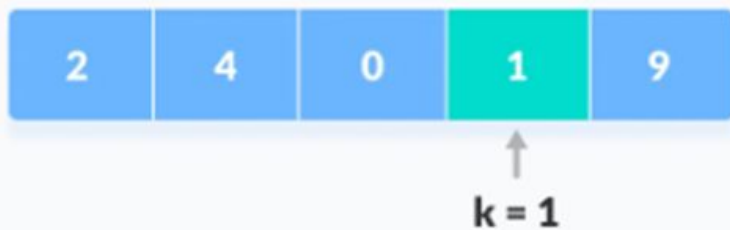
$k \neq 4$



$k \neq 0$

Bandingkan dengan setiap elemen

2. Jika `x == k`, kembalikan index.



Elemen ditemukan

3. Lain, kembali `tidak ditemukan`.

Contoh Program

```
// Linear Search in C++
```

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

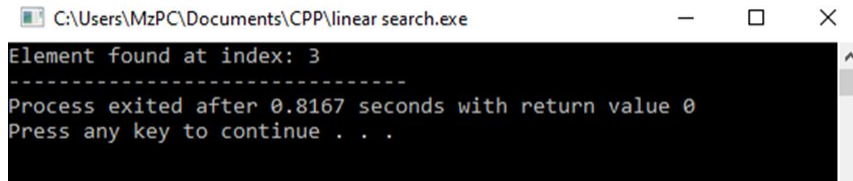
```
int search(int array[], int n, int x) {
```

```
    // Going through array sequentially
    for (int i = 0; i < n; i++)
        if (array[i] == x)
            return i;
    return -1;
}
```

```
int main() {
    int array[] = {2, 4, 0, 1, 9};
    int x = 1;
    int n = sizeof(array) / sizeof(array[0]);
```

```
    int result = search(array, n, x);
```

```
    (result == -1) ? cout << "Element not found" :
    cout << "Element found at index: " << result;
}
```



```
C:\Users\MzPC\Documents\CPP\linear search.exe
Element found at index: 3
-----
Process exited after 0.8167 seconds with return value 0
Press any key to continue . . .
```

Algoritma Binary Search

Pencarian Biner adalah algoritma pencarian untuk menemukan posisi elemen dalam array yang diurutkan. Dalam pendekatan ini, elemen selalu dicari di tengah sebagian dari array.

Pencarian biner hanya dapat diimplementasikan pada daftar item yang diurutkan. Jika elemen belum diurutkan, kita perlu mengurutkannya terlebih dahulu.

Cara Kerja

Algoritma Pencarian Biner dapat diimplementasikan dalam dua cara yang dibahas di bawah ini.

- Metode Iteratif
- Metode Rekursif

Metode rekursif mengikuti pendekatan membagi dan menaklukkan .

<https://www.programiz.com/dsa/binary-search>

Contoh Program

// Binary Search in C++

```
#include <iostream>
```

```
using namespace std;
```

```
int binarySearch(int array[], int x, int low, int high) {
```

```
    // Repeat until the pointers low and high meet each other
```

```
    while (low <= high) {
```

```
        int mid = low + (high - low) / 2;
```

```
        if (array[mid] == x)
```

```
            return mid;
```

```
        if (array[mid] < x)
```

```
            low = mid + 1;
```

```
        else
```

```
            high = mid - 1;
```

```
    }
```

```
    return -1;
```

```
}
```

```
int main(void) {
```

```
    int array[] = {3, 4, 5, 6, 7, 8, 9};
```

```
    int x = 4;
```

```
    int n = sizeof(array) / sizeof(array[0]);
```

```
    int result = binarySearch(array, x, 0, n - 1);
```

```
    if (result == -1)
```

```
        printf("Not found");
```

```
    else
```

```
        printf("Element is found at index %d", result);
```

```
}
```

 C:\Users\MzPC\Documents\CPP\ (Metode Iteratif).exe

Element is found at index 1

Process exited after 2.482 seconds with return value 0

Press any key to continue . . .