

RESUME MATER

SEARcing ARRAY

Mata Kuliah : Struktur Data
Dosen Pengampu : Andi Moch Januriana, ST., M.Kom



NAME : Mustopa
NIM : 3337220023
KELAS : C

Program Studi Informatika
Fakultas Teknik
Universitas Sultan Ageng Tirtayasa

Pengertian array

- array adalah sekumpulan elemen elemen yang di beri index awal 0
- array dapat berupa 1 dimensi dan 2 dimensi bahkan sampai n di mensi

Array 1 dimensi

- hanya mempunyai 1 baris

contoh

[0]	[1]	[2]	[3]	[4]	← index
10	15	30	8	9	← elemen

Array 2 dimensi

- mempunyai baris dan kolom
- pengaksesannya sama seperti matriks
- dimulai dari index [0][0]
- [0] pertama untuk menyatakan baris
- Dan [0] kedua untuk kolom

ILUSTRASI

Int A [3][4];

Materi Dosen	Kolom 0	Kolom 1	Kolom 2	Kolom 3
Baris 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
Baris 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
Baris 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

SEARCHING ARRAY

- METODE METODE PENCARIAN DI ELEMEN ELEMEN ARRAY

Sequential Search

Adalah metode search array yang mengecek semua elemen array dari awal sampai akhir

Dan masih banyak lagi metode metode search array

Soal-soal



- Cari tahu tentang cara penggunaan dan teknologi dari website-website pencari (search engine) yang ada di Internet!
- Cari tahu tentang Fibonacci Search!

CARA KERJA SEARCH ENGINE

Secara umum, cara kerja search engine adalah sebagai berikut:

1. Crawling: Search engine mengirimkan program komputer yang disebut spider atau crawler ke situs web untuk mengekstrak informasi dari halaman web dan mengumpulkannya ke dalam basis data mereka.

2.Indexing: Informasi yang dikumpulkan oleh spider kemudian diindeks dan disimpan dalam basis data search engine untuk kemudian diambil ketika seseorang melakukan pencarian.

3.Ranking: Ketika seseorang melakukan pencarian, search engine membandingkan query pencarian dengan indeks yang telah disimpan dan menampilkan hasil pencarian yang relevan dengan urutan yang berdasarkan algoritma pencarian mereka

4.Penilaian hasil pencarian: Setelah hasil pencarian ditampilkan, search engine melakukan penilaian lebih lanjut dengan mengumpulkan informasi tentang bagaimana pengguna berinteraksi dengan hasil pencarian. Hal ini dapat mempengaruhi urutan hasil pencarian di masa mendatang.

Beberapa faktor yang mempengaruhi ranking dalam hasil pencarian antara lain relevansi konten, kualitas konten, struktur website, kecepatan website, popularitas website, dan faktor lain yang berhubungan dengan SEO (Search Engine Optimization).

DERET FIBONACI

Adalah deret angka yang di tambah sama angka sebelumnya atau

Contoh

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,.....N

FIBONACI SEARCH

Fibonacci search (atau Fibonacci search technique) adalah salah satu algoritma pencarian yang digunakan untuk mencari nilai dalam suatu himpunan data terurut. Algoritma ini didasarkan pada deret Fibonacci, yaitu deret bilangan bulat yang setiap elemennya adalah jumlah dari dua elemen sebelumnya, dengan dua elemen pertama bernilai 0 dan 1.

Pada dasarnya, algoritma Fibonacci search mengurangi jumlah elemen yang perlu dibandingkan dalam setiap iterasi. Algoritma ini membagi himpunan data menjadi dua bagian, di mana rasio antara jumlah elemen pada bagian-bagian tersebut adalah deret Fibonacci. Kemudian, algoritma ini memeriksa elemen di indeks yang sesuai dengan angka Fibonacci terdekat di bawah atau sama dengan jumlah elemen yang ingin dicari. Jika nilai elemen yang diperiksa lebih besar dari nilai yang ingin dicari, algoritma ini membatasi himpunan data ke subhimpunan yang lebih kecil di sebelah kiri. Jika nilai elemen yang diperiksa lebih kecil dari nilai yang ingin dicari, algoritma ini membatasi himpunan data ke subhimpunan yang lebih kecil di sebelah kanan. Proses ini terus berlanjut sampai elemen yang dicari ditemukan atau himpunan data yang tersisa hanya terdiri dari satu elemen.

Keuntungan dari algoritma Fibonacci search adalah kecepatan pencariannya yang relatif cepat, dan pengurangan jumlah elemen yang harus dibandingkan dalam setiap iterasi. Namun, algoritma ini memiliki beberapa kelemahan, seperti kompleksitas implementasi

yang relatif tinggi dan sulit untuk diterapkan pada himpunan data yang tidak terurut

CONTOH CODE DI C++

```
1 #include <iostream>
2 using namespace std;
3
4 int fibonacciSearch(int arr[], int n, int x) {
5     int fib2 = 0; // elemen ke-2 dalam deret Fibonacci
6     int fib1 = 1; // elemen ke-1 dalam deret Fibonacci
7     int fib = fib2 + fib1; // elemen ke-3 dalam deret Fibonacci
8
9     // mencari angka Fibonacci terdekat di bawah atau sama dengan n
10    while (fib < n) {
11        fib2 = fib1;
12        fib1 = fib;
13        fib = fib2 + fib1;
14    }
15
16    int offset = -1;
17
18    // membatasi himpunan data sampai hanya terdiri dari satu elemen
19    while (fib > 1) {
20        int i = min(offset + fib2, n - 1);
21
22        if (arr[i] < x) {
23            fib = fib1;
24            fib1 = fib2;
25            fib2 = fib - fib1;
26            offset = i;
27        }
28        else if (arr[i] > x) {
29            fib = fib2;
30            fib1 = fib1 - fib2;
31            fib2 = fib - fib1;
32        }
33        else if (arr[i] == x) {
34            return i;
35        }
36    }
37
38    if (fib1 && arr[offset + 1] == x) {
39        return offset + 1;
40    }
41    return -1;
42 }
43
44 int main() {
45     int arr[] = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
46     int n = sizeof(arr) / sizeof(arr[0]);
47     int x;
48     cout<<"masukan nilai yang di cari dalam array: ";
49     cin>>x;
50
51     int index = fibonacciSearch(arr, n, x);
52
53     if (index == -1) {
54         cout << "Angka " << x << " tidak ditemukan dalam himpunan data" << endl;
55     }
56     else {
57         cout << "Angka " << x << " ditemukan pada indeks ke-" << index << endl;
58     }
59     return 0;
60 }
```

OUTPUT

```
16 int offset = -1;
masukan nilai yang di cari dalam array: 9
Angka 9 ditemukan pada indeks ke-4

...Program finished with exit code 0
Press ENTER to exit console.
```