

Nama:India

Nim :D0424320

Jenis-Jenis Algoritma

Algoritma adalah serangkaian langkah sistematis untuk menyelesaikan masalah atau mencapai tujuan. Karakteristik utama algoritma meliputi langkah-langkah yang jelas, adanya input dan output, dan berakhir setelah sejumlah langkah tertentu. Contoh sederhana adalah algoritma untuk menjumlahkan dua angka: ambil input, hitung total, dan tampilkan hasil. Algoritma penting dalam pemrograman dan pengolahan data.

Seorang matematikawan dan ilmuwan komputer bernama Dr. Christoph Koutschan mengatakan bahwa setidaknya ada 32 algoritma dalam ilmu komputer. Namun jika dilihat dari fungsinya, hanya ada enam algoritma dasar, yaitu:

1. Rekursi

Sebuah algoritma rekursi adalah sesuatu yang akan memanggil dirinya sendiri berulang-ulang sehingga masalah dapat diselesaikan dengan benar. Berikut adalah beberapa kode yang dapat menemukan faktorial menggunakan algoritma rekursi.

```
Fact(y)
```

```
If y is 0
```

```
return 1
```

```
return (y*Fact(y-1)) /* this is where the recursion happens*/
```

2. Divide and Conquer

Divide and Conquer akan membagi masalah besar menjadi banyak menjadi masalah kecil. Jenis algoritma ini sendiri sebenarnya terdiri dari dua bagian utama, yaitu:

Memecah masalah menjadi submasalah yang lebih independen dan lebih kecil dari masalah lain yang sejenis.

Memecahkan masalah asli setelah dapat menyelesaikan masalah yang lebih kecil secara terpisah.

Berikut ini adalah contoh kode semu dari algoritma divide and conquer:

```
MergeSorting(ar[], l, r)
```

```
If r > l
```

Find the mid-point to divide the given array into two halves:

```
middle m = (l+r)/2
```

Call mergeSorting for the first half:

```
Call mergeSorting(ar, l, m)
```

Call mergeSorting for the second half:

```
Call mergeSorting(ar, m+1, r)
```

Merge the halves sorted in step 2 and 3:

Call merge(ar, l, m, r)

3. Dynamic Programming

Dynamic Programming akan bekerja dengan mengingat hasil dari proses masa lalu dan juga menggunakannya untuk menemukan hasil baru. Dari penjelasan diatas, berarti bahwa Dynamic Programming memecahkan masalah kompleks dengan memecahnya menjadi banyak submasalah sederhana, kemudian menyelesaikannya satu per satu, lalu menyimpannya untuk digunakan di masa mendatang.

Contoh Dynamic Programming adalah terdiri dari urutan Fibonacci, berikut adalah contohnya:

Fibonacci(N) = 0 (for n=0)

= 0 (for n=1)

= Fibonacci(N-1) + Fibonacci(N-2)

4. Greedy

Tipe ini dapat digunakan untuk menyelesaikan masalah optimasi. Dalam algoritma ini, kita akan menemukan solusi yang lebih optimal secara lokal tanpa mengkhawatirkan konsekuensi yang akan terjadi di masa depan dan kita dapat menemukan solusi yang lebih optimal secara global.

Namun, cara ini tidak menjamin Anda bisa mendapatkan solusi yang optimal. Ada lima komponen yang terdapat dalam algoritma ini, yaitu:

Himpunan kandidat yang solusinya akan dicari kemudian.

Fungsi seleksi terakhir akan membantu memilih kandidat terbaik.

Fitur kelayakan dapat membantu Anda mengidentifikasi kandidat mana yang dapat digunakan untuk menemukan solusi.

Fungsi tujuan mampu memberikan nilai ke solusi yang mungkin atau solusi parsial.

Fungsi solusi dapat memberitahu Anda tentang waktu yang diperlukan untuk menemukan solusi untuk masalah tersebut.

5. Brute Force

Konsep dari algoritma ini sebenarnya sangat sederhana. Selama proses ini, brute force mengintegrasikan semua solusi yang mungkin untuk menemukan satu atau lebih solusi yang mungkin untuk memecahkan masalah.

Berikut adalah contoh pencarian sekuensial yang dilakukan menggunakan brute force:

Algorithm S_Search (A[0..n], X)

A[n] \leftarrow X

i \leftarrow 0

While A [i] \neq X do

```
i ← i + 1  
if i < n return i  
else return -1
```

6. Algoritma Backtracking

Backtracking adalah teknik yang dapat memecahkan masalah yang berbeda secara rekursi dan mencoba mereka untuk menemukan solusi dengan memecahkan satu bagian dari masalah pada waktu yang sama. Jika solusi gagal, kita dapat menghapusnya dan kembali mencari solusi lain.

Artinya algoritma ini akan menyelesaikan sub masalah dan jika gagal maka algoritma akan membatalkan langkah terakhir dan memulai kembali untuk mencari solusi dari masalah tersebut.

Menurut Donald E. Knuth, algoritma harus memiliki lima karakteristik penting yang saling berhubungan. Kriteria untuk algoritma ini meliputi:

1. Finiteness (Keterbatasan)

Algoritma harus berhenti setelah mengambil sejumlah langkah yang terbatas, yaitu ada tujuan akhir yang tercapai, sehingga program akan berhenti ketika tujuan akhir telah tercapai. Program yang tidak pernah berhenti menunjukkan bahwa program tersebut mengandung algoritma yang salah.

2. Definiteness (Kepastian)

Setiap langkah harus didefinisikan dengan tepat dan tidak ambigu. Ada instruksi yang jelas dan tidak ambigu, sehingga tidak ada kesalahan dalam menghasilkan output.

3. Input (Masukan)

Input ini adalah masalah yang diketahui dan solusi akan diselidiki. Algoritma ini tidak memiliki atau lebih input, yang merupakan jumlah yang disediakan untuk algoritma untuk diproses.

4. Output (Keluar)

Algoritma tidak memiliki atau lebih nilai output. Output ini tentunya harus menjadi solusi atau solusi dari suatu masalah. Output dapat berupa pesan atau kuantitas yang terkait dengan input.

5. Effectiveness (keefektivitasan)

Algoritma harus efisien, setiap urutan atau langkah harus sesederhana mungkin, sehingga dapat diimplementasikan dalam waktu yang wajar.

Kesimpulan

Algoritma merupakan inti dari pemrograman dan komputasi modern. Masing-masing jenis algoritma memiliki kekuatan dan kelemahan tergantung pada masalah yang dihadapi. Dengan memahami berbagai jenis algoritma seperti brute force, divide and conquer, greedy, dynamic programming, dan backtracking, kita dapat memilih algoritma yang paling sesuai untuk menyelesaikan masalah secara efektif dan efisien.