Joseph Janone Tiwouw Informatika

Reading Report on Searching Algorithms……

-Refrensi pertama (<https://study.com/academy/lesson/string-searching-algorithms-methods-types.html>)

Dalam kehidupan kita, secara tidak sadar maupun saat kita sadar, kita menggunakan searching algorithm. Misalnya, setiap pencarian yang kita masukkan ke dalam mesin pencari diperlakukan sebagai substring dan mesin pencari menggunakan searching algorithm untuk memberi Anda hasil. Saat Anda memasukkan string, mesin pencari mencari sejumlah besar data di internet dan memberi Anda hasil pencarian.

-Refrensi kedua (<https://en.wikipedia.org/wiki/String-searching_algorithm>)

Dalam ilmu komputer, algoritma pencarian string, kadang-kadang disebut algoritma pencocokan string, adalah kelas penting dari algoritma string yang mencoba menemukan tempat di mana satu atau beberapa string (juga disebut pola) ditemukan dalam string atau teks yang lebih besar. Contoh dasar pencarian string adalah ketika pola dan teks yang dicari adalah array dari elemen alfabet (kumpulan hingga) . mungkin alfabet bahasa manusia, misalnya, huruf A sampai Z dan aplikasi lain dapat menggunakan alfabet biner (Σ = {0,1}) atau alfabet DNA (Σ = {A,C,G,T}) dalam bioinformatika. Dalam prakteknya, metode algoritma pencarian string yang layak dapat dipengaruhi oleh pengkodean string. Secara khusus, jika pengkodean lebar variabel sedang digunakan, maka mungkin lebih lambat untuk menemukan karakter ke-N, mungkin memerlukan waktu yang sebanding dengan N. Hal ini dapat secara signifikan memperlambat beberapa algoritme pencarian. Salah satu dari banyak solusi yang mungkin adalah mencari urutan unit kode sebagai gantinya, tetapi melakukan hal itu dapat menghasilkan kecocokan yang salah kecuali pengkodean dirancang khusus untuk menghindarinya.

-Refrensi ketiga (<https://p2k.undaris.ac.id/IT/2-3072-2969/String-searching-algorithm_14953_p2k-undaris.html>)

Algoritma pencarian lebih cepat didasarkan pada pra-pemrosesan teks. Setelah membangun indeks substring, misalnya pohon sufiks atau larik sufiks, kemunculan suatu pola dapat ditemukan dengan cepat. Sebagai contoh, pohon sufiks dapat dibangun dalam waktu \Theta(n), dan semua z kemunculan suatu pola dapat ditemukan dalam waktu O(m+z) (jika ukuran alfabet dipandang sebagai konstanta).

-Refrensi keempat (<https://www.geeksforgeeks.org/searching-algorithms/>)

Algoritma Pencarian dirancang untuk memeriksa elemen atau mengambil elemen dari struktur data mana pun yang menyimpannya. Berdasarkan jenis operasi pencarian, algoritma ini umumnya diklasifikasikan menjadi dua kategori:

Pencarian Berurutan: Dalam hal ini, daftar atau larik dilalui secara berurutan dan setiap elemen diperiksa. Misalnya: Pencarian Linier.

Pencarian Interval: Algoritma ini dirancang khusus untuk pencarian dalam struktur data yang diurutkan. Jenis algoritma pencarian ini jauh lebih efisien daripada Pencarian Linear karena mereka berulang kali menargetkan pusat struktur pencarian dan membagi ruang pencarian menjadi dua. Misalnya: Pencarian Biner.

-Refrensi kelima (<https://en.wikipedia.org/wiki/Search_algorithm>)

Algoritma pencarian dapat diklasifikasikan berdasarkan mekanisme pencariannya menjadi 3 jenis algoritma: linier, biner, dan hashing. Algoritme pencarian linier memeriksa setiap catatan untuk yang terkait dengan kunci target secara linier. Pencarian biner, atau setengah interval, berulang kali menargetkan pusat struktur pencarian dan membagi ruang pencarian menjadi dua. Algoritme pencarian perbandingan meningkatkan pencarian linier dengan menghilangkan catatan secara berurutan berdasarkan perbandingan kunci sampai catatan target ditemukan, dan dapat diterapkan pada struktur data dengan urutan yang ditentukan. Algoritma pencarian digital bekerja berdasarkan sifat-sifat digit dalam struktur data yang menggunakan kunci numerik. Terakhir, hashing langsung memetakan kunci ke record berdasarkan fungsi hash. Algoritma sering dievaluasi berdasarkan kompleksitas komputasinya, atau waktu berjalan teoritis maksimum. Fungsi pencarian biner, misalnya, memiliki kompleksitas maksimum O(log n), atau waktu logaritmik. Ini berarti bahwa jumlah operasi maksimum yang diperlukan untuk menemukan target pencarian adalah fungsi logaritmik dari ukuran ruang pencarian.