

Soal Latihan Brute Force

IF2211 Strategi Algoritma

2023

UTS 2022

Diberikan sebuah larik yang berisi elemen biner (0 atau 1). Elemen-elemen larik sudah **terurut** menaik (dari kecil ke besar). Kita akan menghitung jumlah bit 1 di dalam larik tersebut. Contoh: $A = [0, 0, 0, 1, 1, 1, 1]$, jumlah bit 1 adalah 5. Jika diselesaikan dengan algoritma brute force, jelaskan caranya atau langkah-langkahnya, lalu tentukan kompleksitas waktu asimptotiknya dalam notasi O besar.

Jawaban:

- Traversal larik sampai ketemu elemen 1 pertama. Karena larik sudah terurut, maka hitung jumlah 1 dari posisi tersebut sampai posisi terakhir.
- Kompleksitas waktu algoritmanya adalah $O(n)$ karena:
 - traversal larik dengan mencari elemen 1 pertama: $O(n)$
 - menghitung banyaknya 1 dari posisi 1 pertama sampai terakhir: $O(n)$
 - total waktu: $O(n) + O(n) = O(n)$

UTS 2022

Seorang mahasiswa IF tingkat 4 mengalokasikan waktu m jam untuk mengerjakan proyek programming. Terdapat n proyek ($p_1..p_n$) yang mungkin diterima dengan mempertimbangkan estimasi waktu kerja ($t_1..t_n$) dalam jam, dan honor yang akan diterima ($s_1..s_n$) dalam ratusan ribu. Bantulah mahasiswa ini memilih proyek yang akan dikerjakan untuk memaksimumkan honor yang akan diterima.

Jika diselesaikan dengan exhaustive search, jelaskanlah langkah-langkah yang dilakukan di dalam exhaustive search. Lengkapilah penjelasan dengan jumlah kandidat solusi yang akan dievaluasi, dan tentukan kompleksitas waktu asimptotiknya dalam notasi O besar.

Jawaban:

Ini adalah persoalan knapsack problem. Langkah-langkah dalam exhaustive search:

- i) enumerasi kandidat solusi dengan sistematis. Jumlah kandidat solusi adalah sebanyak subset dari n buah proyek: 2^n buah subset
- ii) evaluasi setiap kandidat solusi dengan menghitung apakah total waktu yang dibutuhkan untuk mengerjakan proyek tidak melebihi m . Jika tidak melebihi m , hitung total honor. Kebutuhan waktu: $O(n)$
- iii) umumkan solusi terbaik di akhir pencarian.

Kompleksitas waktu $O(n \cdot 2^n)$.

UTS 2021

- Misalkan terdapat sebuah larik $a[1..n]$ dengan n elemen bilangan bulat. Kita ingin menghitung $F = \sum_{i=1}^n i * a[i]$ sedemikian sehingga F bernilai maksimum. Jika diselesaikan secara *brute force/exhaustive search*, bagaimana caranya, dan berapa perkiraan kompleksitas waktunya (dalam notasi Big-Oh)?

Jawaban

- Urutkan larik dari kecil ke besar
- Inisialisasi $F = 0$
- Traversal dari $k = 1$ sampai n ,
$$F = F + k * a[k]$$

Kompleksitas waktu = waktu pengurutan + waktu traversal
$$= O(n^2) + O(n)$$
$$= O(n^2)$$

Adakah algoritam brute force yang lebih baik dari yang di atas?

UTS 2016

Misalkan anda diberikan sebuah larik bilangan bulat yang terurut. Setiap nilai muncul dua kali, kecuali sebuah nilai tertentu yang hanya muncul sekali. Tugas anda adalah mencari nilai integer yang muncul hanya sekali.

Contoh larik:

- (i) 1, 1, 2, 2, **3**, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8
- (ii) 10, 10, 17, 17, 18, 18, 19, 19, 21, 21, **23**
- (iii) **1**, 3, 3, 5, 5, 7, 7, 8, 8, 9, 9, 10, 10

(angka yang dicetak tebal adalah nilai yang hanya muncul sekali)

Jika diselesaikan dengan algoritma Brute Force, bagaimana caranya? (jawaban bukan dalam *pseudo-code*). Berapa kompleksitasnya dalam notasi O-besar?