
EXERCISE 3.2: ALGORITMA BRUTE-FORCEdikerjakan saat perkuliahan tatap muka pertemuan 3

- (Sub-array terbesar)* Diberikan array bilangan bulat a_1, a_2, \dots, a_n . Anda diminta untuk mencari sub-array yang memiliki jumlah maksimal. Nilai maksimum sub-urutan adalah nol jika semua elemen array negatif. Sebagai contoh: array $[-2, 11, -4, 13, -5, 2, -1, 3]$ memiliki nilai maksimum sub-array 20, yaitu dari elemen ke-2 hingga elemen ke-4 (yaitu $[11, -4, 13]$, jumlah totalnya adalah $11 + (-4) + (13) = 20$).
- (Banyak pemberhentian minimum)* Misalkan kita menempuh jarak sejauh 100 km, mulai dari titik 0 dan berakhir di kilometer 100. Terdapat pom bensin di sepanjang jalan dengan jarak 10, 25, 30, 40, 50, 75 dan 80 km dari titik awal. Tangki bensin awalnya hanya cukup untuk menempuh jarak 30 km. Bagaimana kita sampai ke tempat pemberhentian sehingga kita berhenti sesedikit mungkin?
- (Gadget testing)* Suatu perusahaan sedang melakukan uji coba untuk menentukan rantai tertinggi dari kantor pusatnya yang berlantai n , yang memungkinkan sebuah gadget dapat jatuh tanpa mengalami kerusakan. Perusahaan tersebut memiliki dua gadget identik untuk bereksperimen. Jika salah satunya rusak, maka gadget tersebut tidak dapat diperbaiki, dan eksperimen harus diselesaikan dengan gadget yang tersisa. (Catatan: kompleksitas waktu dihitung berdasarkan banyaknya percobaan yang dilakukan.)
 - Rancang sebuah algoritma dengan pendekatan brute-force untuk masalah ini. Jelaskan algoritmanya, kemudian tulis pseudocode untuk algoritma tersebut, dan tentukan kompleksitas waktunya!
 - Modifikasi algoritma tersebut untuk meningkatkan efisiensinya. (*Hint*: kita bisa membuat algoritma dengan kompleksitas waktu *worst-case* $\mathcal{O}(\sqrt{n})$.)
- (Titik-titik terdekat)* Misalkan $x_1 < x_2 < \dots < x_n$ adalah bilangan real yang mewakili koordinat n desa yang terletak di sepanjang jalan lurus. Kantor pos perlu dibangun di salah satu desa ini.
 - Rancang algoritma yang efisien untuk menemukan lokasi kantor pos yang meminimalkan jarak rata-rata antara desa dan kantor pos.
 - Rancang algoritma yang efisien untuk menemukan lokasi kantor pos yang meminimalkan jarak maksimum dari desa ke kantor pos.
- (Permasalahan partisi)* **Permasalahan partisi** didefinisikan sebagai berikut.

Diberikan n bilangan bulat positif, partisi menjadi dua himpunan bagian yang terpisah dengan jumlah elemen yang sama. (Perhatikan bahwa masalah ini tidak selalu memiliki solusi.) Rancang algoritma exhaustive search untuk masalah ini, dengan meminimalkan jumlah himpunan bagian yang perlu dihasilkan oleh algoritma.

Jelaskan algoritmanya, tuliskan pseudocode-nya, dan hitung kompleksitas waktu terburuknya!