

Desain dan Analisis Algoritma

Semester Ganjil 2023/2024

Tugas Eksperimen 2 - Algorithm Design Techniques

Deadline: Rabu, 6 Desember 2023 pukul 08:00 WIB

Petunjuk Pengerjaan:

- Berkas TE Anda diketik dengan ukuran kertas A4 lalu dikonversikan ke PDF (bukan .doc{x}). Mengumpulkan selain tipe file PDF dikenakan penalti sebesar 5 poin.
- Format penamaan berkas TE2 NPM Nama.pdf.
Contoh: TE2 2106123456 John Doe.pdf.
Penamaan berkas yang tidak sesuai dikenakan penalti sebesar 5 poin.
- Tuliskan Nama, NPM, dan Kode Asdos Anda pada bagian atas halaman pertama laporan TE Anda. Kode Asdos yang tidak sesuai dikenakan penalti sebesar 5 poin.
- Awali berkas TE Anda dengan pernyataan “Dengan ini saya menyatakan bahwa TE ini adalah hasil pekerjaan saya sendiri” disertai tanda tangan pada halaman pertama berkas TE Anda.
Tanpa pernyataan ini, TE Anda tidak akan diperiksa.
- Jika ada, tuliskan Nama Kolaborator pada berkas TE Anda. Perhatikan bahwa walaupun Anda sudah menuliskan nama kolaborator, bukan berarti jawaban Anda boleh sama persis dengan kolaborator Anda. TE adalah tugas individu, bukan tugas kelompok. Pastikan kolaborasi hanya pada sebatas ide pengerjaan, bukan ketika menulis jawaban.
- Tidak diperbolehkan untuk melakukan plagiasi dalam penulisan laporan. Setiap informasi, teks, atau ide yang berasal dari sumber eksternal harus diberi referensi dengan jelas.
- Pelanggaran peraturan kejujuran akademis akan diproses sesuai peraturan yang sudah dijelaskan di BRP.
- **Tidak ada toleransi terhadap keterlambatan pengumpulan TE.**

Pada Tugas Eksperimen 2 ini, Anda diminta untuk membandingkan performa dari dua teknik desain algoritma pada salah satu problem dibawah ini. Pemilihan problem sesuai dengan pembagian asdos di SCellE.

Problem 1 (Vertex Cover on Trees): Diberikan sebuah tree, yaitu graf $G = (V, E)$, dengan V adalah himpunan simpul (*vertex*) dan E adalah himpunan sisi (*edge*), yang terkoneksi dan tidak memiliki loop. Sebuah **vertex cover** dari graf tersebut adalah himpunan simpul $V' \subseteq V$ sedemikian sehingga setiap sisi dalam graf memiliki setidaknya satu ujungnya yang terhubung ke salah satu simpul dalam V' . Tujuan dari masalah Vertex Cover adalah untuk menemukan vertex cover dengan jumlah simpul minimum.

Algoritma: Dynamic Programming [URL] vs Branch and Bound [URL]

Problem 2 (Hamiltonian Path): Diberikan sebuah graf planar tak berarah $G = (V, E)$, di mana V adalah himpunan simpul (*vertex*) dan E adalah himpunan sisi (*edge*). Sebuah **Hamiltonian Path** dari graf tersebut adalah urutan simpul v_1, v_2, \dots, v_n sedemikian sehingga (1). semua $v \in V$ muncul tepat sekali dalam urutan tersebut, (2). untuk setiap $1 \leq i < n$, terdapat sisi $(v_i, v_{i+1}) \in E$.

Algoritma: Dynamic Programming [URL] vs Backtracking [URL]

Problem 3 (Weighted Set Cover): Diberikan sebuah himpunan universal U yang terdiri dari n elemen, dan sebuah himpunan $S = \{S_1, S_2, \dots, S_m\}$ dengan setiap $S_i \subseteq U$. Selain itu, terdapat fungsi $w : 2^U \rightarrow \mathbb{R}$ yang memetakan setiap subset dari U dengan suatu nilai *weight* bilangan riil. Carilah subset $S' \subseteq S$ sedemikian hingga $\bigcup_{S'' \in S'} S'' = U$, dan S' merupakan subset dengan total weight minimum.

Algoritma: Greedy [URL] vs Branch and Bound [URL]

Problem 4 (0/1 Unbounded Knapsack): Diberikan n buah barang, masing-masing dengan berat w_i dan harga v_i yang tak hingga banyaknya, serta sebuah ransel dengan kapasitas berat maksimum W . Tentukan cara untuk memilih barang-barang tersebut sedemikian hingga jumlah beratnya kurang dari atau sama dengan W dan jumlah harganya dimaksimalkan.

Algoritma: Dynamic Programming [URL] vs Branch and Bound [URL]

Problem 5 (Set Partition): Diberikan sebuah himpunan $S = \{s_1, s_2, \dots, s_n\}$ berisi n buah bilangan bulat positif. Bagi S menjadi dua partisi S_1 dan S_2 sedemikian hingga $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \emptyset$, dan $\sum_{s \in S_1} s = \sum_{t \in S_2} t$. Artinya, jumlah semua elemen di S_1 sama dengan jumlah semua elemen di S_2 .

Algoritma: Dynamic Programming [URL] vs Branch and Bound [URL]

Tugas Anda adalah:

1. Implementasikan kembali algoritma-algoritma tersebut dalam bahasa pemrograman Python / Java sedemikian hingga bisa diperbandingkan.
2. Analisis perbandingan penggunaan memori dan waktu eksekusi dari kedua algoritma tersebut. Untuk ini, Anda perlu membuat sebuah algoritma yang *men-generate* dataset dengan variasi ukuran: Kecil, Sedang, Besar.

<i>Problem/ Kode Asdos</i>	Ukuran Kecil	Ukuran Sedang	Ukuran Besar
1.	10^4 vertex (DP) BnB: Menggunakan dataset yang sama (10^4), tetapi ambil hanya 100 vertex pertama	10^5 vertex (DP) BnB: Menggunakan dataset yang sama (10^5), tetapi ambil hanya 300 vertex pertama	10^6 vertex (DP) BnB: Menggunakan dataset yang sama (10^6), tetapi ambil hanya 900 vertex pertama
2.	16 vertex	18 vertex	20 vertex
3.	20 elemen	200 elemen	2.000 elemen
4.	100 barang	1.000 barang	10.000 barang
5.	10 elemen	40 elemen	80 elemen

3. Analisis perbandingan hasil *running time* dengan kompleksitas dari Algoritma tersebut secara teori (dalam notasi asimtotik). Jelaskan secara singkat kompleksitas algoritma yang kalian gunakan.
4. Tuliskan laporan Tugas Eksperimen ini dalam 2-3 halaman (tidak termasuk *pseudocode*).
Format: Ukuran Kertas A4, Font Times New Roman 12, line space: single.
Isinya mencakup:
 - Deskripsi singkat algoritma baru tersebut beserta contoh penerapannya (dengan input buatan sendiri).
 - Pseudocode serta tautan Github/Gitlab yang berisi implementasi dan dataset yang digunakan.
 - Hasil eksperimen dan analisis.
 - Kesimpulan.