

Desain dan Analisis Algoritma

Semester Ganjil 2023/2024

Tugas Eksperimen 1 - Sorting Algorithms

Deadline: Senin, 6 November 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 TE1 NPM Nama.pdf.
Contoh: TE1 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 1 ini, Anda diminta untuk membandingkan satu algoritma sorting yang sudah pernah dipelajari di kelas dengan satu algoritma sorting baru, sebagai berikut.

Kode Asdos	Algoritma 1	Algoritma 2
1.	Bidirectional-Conditional-Insertion-Sort [1]	Counting Sort
2.	Peeksort [2]	Radix Sort using Counting Sort
3.	Two-Pivot-Block-Quicksort [3]	Merge Sort
4.	Randomized Shell Sort [4]	Max-Heap Sort
5.	Clustered-Binary-Insertion-Sort [5]	Randomized Quick Sort

Tugas Anda adalah:

1. Implementasikan algoritma-algoritma sorting tersebut dalam bahasa pemrograman Python / Java. Setiap mahasiswa mengerjakan satu pasangan algoritma menurut pembagian asdos di SCellE.
2. Analisis perbandingan penggunaan memori dan waktu eksekusi (dalam milliseconds [ms]) dari kedua algoritma tersebut. Untuk ini, Anda perlu membuat sebuah algoritma yang *men-generate* dataset dengan variasi:

Kode Asdos	Ukuran Kecil	Ukuran Sedang	Ukuran Besar
1.	500 angka	5.000 angka	50.000 angka
2.	1.000 angka	10.000 angka	100.000 angka
3.	2^9 angka	2^{13} angka	2^{16} angka
4.	2^9 angka	2^{13} angka	2^{16} angka
5.	200 angka	2.000 angka	20.000 angka

- Status: *sorted, random, reversed*

Sehingga ada 9 kombinasi untuk dataset.

3. Analisis perbandingan hasil *running time* dengan kompleksitas dari Algoritma tersebut secara teori (dalam notasi asimtotik). Anda tidak perlu menjelaskan secara detail mengenai kompleksitas asimtotik tersebut.
4. Tuliskan laporan Tugas Eksperimen ini dalam 2–3 halaman (tidak termasuk *pseudocode*).
Format: Ukuran Kertas A4, Font Times New Roman 12.
Isinya mencakup:

- Deskripsi singkat algoritma baru tersebut beserta contoh penerapannya (dengan input buatan sendiri).
- Pseudocode serta tautan Github yang berisi implementasi dan dataset yang digunakan.
- Hasil eksperimen dan analisis.
- Kesimpulan.

Catatan untuk setiap Kode Asdos

1. -
2. Dalam paper disebutkan bahwa untuk subproblem kecil, maka bisa digantikan oleh insertion-sort. Namun, karena fokus TE ini adalah melakukan analisis algoritma Peeksort, maka jangan lakukan pergantian tersebut di implementasi Anda.
3. Baca paper baik-baik untuk menemukan informasi mengenai Block Size.
4. Baca paper baik-baik untuk menemukan informasi mengenai *output* yang deterministik dan banyaknya region compare-exchange.
5. -

References

- [1] Adnan Saher Mohammed, Sahin Emrah Amrahov, and Fatih V C, elebi. Bidirectional Conditional Insertion Sort algorithm; An efficient progress on the classical insertion sort. *Future Generation Computer Systems*, 71:102–112, 2017.
- [2] J Ian Munro and Sebastian Wild. Nearly-Optimal Mergesorts: Fast, Practical Sorting Methods That Optimally Adapt to Existing Runs. In *26th Annual European Symposium on Algorithms (ESA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- [3] Martin Aumüller and Nikolaj Hass. Simple and Fast BlockQuicksort using Lomuto’s Partitioning Scheme. In *2019 Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 15–26. SIAM, 2019.
- [4] Michael T Goodrich. Randomized Shellsort: A Simple Data-Oblivious Sorting Algorithm. *Journal of the ACM (JACM)*, 58(6):1–26, 2011.
- [5] Shubham Goel and Ravinder Kumar. Brownian Motus and Clustered Binary Insertion Sort methods: An efficient progress over traditional methods. *Future Generation Computer Systems*, 86:266–280, 2018.