

# Desain dan Analisis Algoritma

Semester Ganjil 2023/2024

## PR 2 - Recurrences, Heapsort, Quicksort

**Deadline:** Senin, 9 Oktober 2023 pukul 08:00 WIB

---

### Petunjuk Pengerjaan:

- Berkas PR Anda harus dibuat dengan cara ditulis tangan di kertas A4 lalu difoto/di scan dan disimpan sebagai satu berkas PDF (bukan di-*zip*). Mengumpulkan selain tipe file PDF dikenakan penalti sebesar 5 poin.
- Format penamaan berkas PR2 NPM Nama.pdf.  
Contoh: PR2 2106123456 John Doe.pdf.  
Penamaan berkas yang tidak sesuai dikenakan penalti sebesar 5 poin.
- Tuliskan Nama, NPM, dan Kode Asdos Anda pada bagian kiri atas setiap halaman pada PR Anda.
- Awali berkas PR Anda dengan pernyataan “Dengan ini saya menyatakan bahwa PR ini adalah hasil pekerjaan saya sendiri” disertai tanda tangan pada halaman pertama berkas PR anda. **Tanpa pernyataan ini, PR Anda tidak akan diperiksa.**
- Jika ada, tuliskan Nama Kolaborator pada berkas PR Anda. Perhatikan bahwa walaupun Anda sudah menuliskan nama kolaborator, bukan berarti jawaban Anda boleh sama persis dengan kolaborator Anda. PR adalah tugas individu, bukan tugas kelompok. Pastikan kolaborasi hanya pada sebatas ide pengerjaan, bukan ketika menulis jawaban.
- Anda harus memberikan penjelasan jawaban pada setiap soal. Bila kurang penjelasan, maka akan dikenakan penalti.
- Anda diperbolehkan menghitung menggunakan kalkulator, namun langkah pengerjaan harus dijelaskan. Tidak boleh menulis nilai akhir saja.
- Pelanggaran peraturan kejujuran akademis akan diproses sesuai peraturan yang sudah dijelaskan di BRP.
- **Tidak ada toleransi terhadap keterlambatan pengumpulan PR.**

1. **Soal 1** ( $6 + 6 + 6 + 6 = 24$  poin)

Gunakan Master's Theorem untuk memperoleh tight asymptotic bound untuk persamaan-persamaan rekurensi berikut (apabila berlaku)

(a)  $T(n) = T(11n/12) + n^2$

(b)  $T(n) = 8T(n/2) + \frac{n^3}{\lg n}$

(c)  $T(n) = 9T(n/3) + n \lg n$

(d)  $T(n) = 16T(n/4) + 2^{\lg(n+1)^2}$

2. **Soal 2** ( $4 + 5 + 5 + 8 + 8 = 30$  poin)

Misal terdapat suatu algoritma yang *running time*-nya dapat dideskripsikan dengan persamaan rekurensi berikut

$$T(n) = 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + \frac{n}{2}$$

- (a) Berapakah level maksimum dari *recurrence tree*-nya?
- (b) Apakah *tree* tersebut penuh sampai ke level maksimumnya? Jika tidak, pada level berapa *tree* tersebut mulai tidak penuh lagi?
- (c) Apakah setiap node dalam level yang sama memiliki *cost* yang sama juga? Tunjukkan *cost* masing-masing node pada level 1, 2, dan 3 beserta total per level-nya.
- (d) Tentukan *upper-bound* dari  $T(n)$  dalam notasi big- $\mathcal{O}$ .
- (e) Gambarkan *tree* yang dibangun berdasarkan pengamatan pada nomor (a) - (d).

3. **Soal 3** ( $6 + 8 + 8 + 8 = 30$  poin)

Ternary heap adalah heap yang sifatnya mirip dengan binary heap tetapi setiap node bisa memiliki tiga anak (left, middle, right).

- (a) Deskripsikan bagaimana merepresentasikan ternary heap dalam sebuah array. Bagaimana menentukan indeks parent dan children dari suatu node di indeks  $i$ ?
- (b) Buatlah algoritma TERNARY-MAX-HEAPIFY( $A, i$ ) untuk ternary heap. Bandingkan kompleksitasnya dengan algoritma MAX-HEAPIFY untuk binary heap.
- (c) Buatlah algoritma DELETE-ELEMENT( $A, x$ ) untuk menghapus elemen  $x$  dari max-ternary-heap  $A$ . Tentukan kompleksitasnya.
- (d) Buatlah algoritma COMBINE-HEAP( $A_1, A_2$ ) untuk menggabungkan dua max-ternary-heap  $A_1$  dan  $A_2$  menjadi max-ternary-heap baru. Tentukan kompleksitasnya.

4. **Soal 4** ( $6 + 4 + 8 = 18$  poin)

Misal kita ubah algoritma partisi untuk quicksort dengan menggunakan algoritma partisi Hoare sebagai berikut

HOARE-PARTITION( $A, p, r$ ):

```
1.  x = p
2.  i = p
3.  j = r + 1
4.  while i < j
5.      repeat j = j - 1 until (A[j] ≤ A[x] or j ≤ p)
6.      repeat i = i + 1 until (A[i] ≥ A[x] or i > r)
7.      if (i < j)
8.          swap A[i] with A[j]
9.  swap A[x] with A[j]
10. return j
```

- (a) Demonstrasikan algoritma HOARE-PARTITION( $A, 1, 8$ ) tersebut untuk memperoleh partisi dari input array  $A = [3, 5, 9, 5, 7, 3, 8, 2]$ .
- (b) Dari demonstrasi tersebut, apakah quicksort dengan partisi Hoare adalah algoritma sorting yang stable? Mengapa?
- (c) Modifikasi quicksort dengan partisi Hoare menggunakan randomisasi. Adakah perbedaan kompleksitas randomized quicksort tersebut dibandingkan menggunakan partisi Lomuto? Apa kegunaan randomisasi pada quicksort?