

# Desain dan Analisis Algoritma

Semester Genap 2023/2024

## PR 2 – Divide and Conquer, Recurrences, and Convolution

Deadline: Rabu, 13 Maret 2024

---

### 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 dan NPM 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 kolab- orator 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.
- Keterlambatan mengumpulkan PR akan dikenai penalti sebesar 30% dari nilai total selama dikumpulkan **maksimal satu jam setelah batas pengumpulan PR.**

**1. Soal 1 (10 + 6 + 4 + 4 + 5 + 7 = 36 Poin)**

Diberikan pseudocode untuk merge sort yang telah Anda pelajari di kelas sebagai berikut:

```
merge_sort(A, p, r):
    if p < r:
        q = [(p+r)/2]
        merge_sort(A, p, q)
        merge_sort(A, q+1, r)
        merge(A, p, q, r)

merge(A, p, q, r):
    n1 = q-p+1
    n2 = r-q
    L <- [1...n1+1] and R <- [1...n2+1]
    for i = 1 to n1:
        L[i] = A[p+i-1]
    for j = 1 to n2:
        R[j] = A[q+j]
    L[n1+1] = ∞
    R[n2+1] = ∞
    i = 1
    j = 1
    for k = p to r:
        if L[i] <= R[j]:
            A[k] = L[i]
            i = i+1
        else:
            A[k] = R[j]
            j = j+1
```

- Lakukan modifikasi pada pseudocode di atas untuk membuat three-way merge sort, yaitu varian merge sort yang membagi array menjadi tiga subarray secara rekursif.
- Berikan visualisasi prosedur three-way merge sort untuk input array  $A = [8, 3, 5, 6, 1, 3, 7, 4, 8, 2]$ , index pertama  $p = 1$ , dan index terakhir  $r = 10$ , serta cara algoritma melakukan merge pada tiga sorted subarray terakhir menjadi sorted array secara detail.
- Pada kasus apa sebuah algoritma sorting bersifat stable? Apakah three-way merge sort yang Anda buat merupakan algoritma sorting yang stable? Berikan alasan Anda.
- Pada kasus apa sebuah algoritma melakukan sorting secara in-place? Apakah three-way merge sort yang Anda buat melakukan sorting secara in-place? Berikan alasan Anda.
- Tuliskan fungsi rekursi untuk running time three-way merge sort. Berapakah running time untuk tahap divide, conquer, dan combine dalam three-way merge sort?
- Tentukan notasi asimptotik big O untuk three-way merge sort menggunakan substitution method. Untuk memudahkan penghitungan, asumsikan  $n$  adalah bilangan kelipatan 3.

## 2. Soal 2 (4 + 4 + 8 + 4 + 5 = 25 Poin)

Diberikan sebuah pseudocode sebagai berikut:

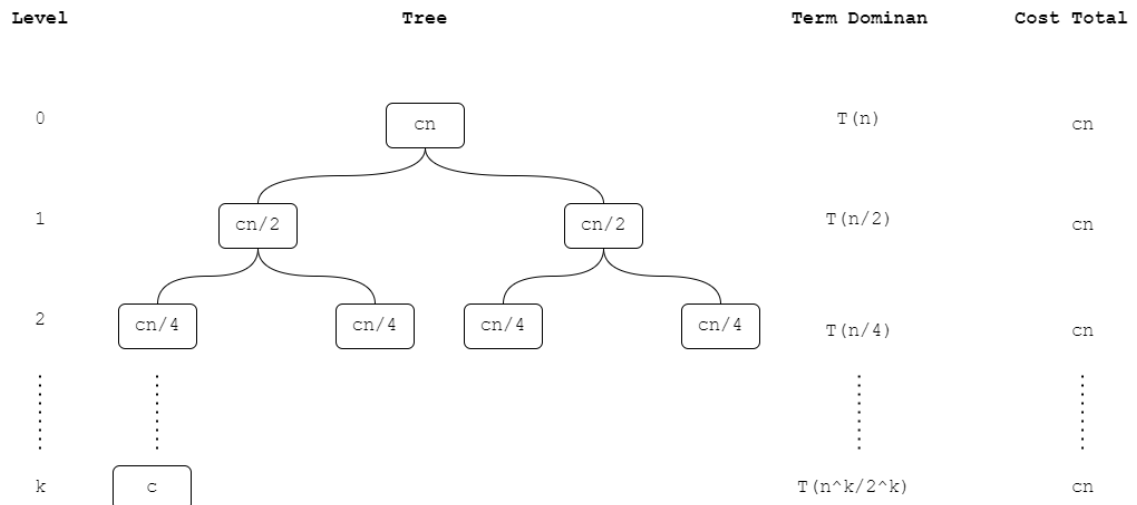
```
mystery(n):
    if (n > 0):
        mystery(n div 4)
        i = 1
        while (i <= n)
            i = i + 1
        mystery((3*n) div 4)
```

Jawablah pertanyaan berikut:

- Tuliskanlah fungsi rekursi dari fungsi mystery tersebut tersebut!
- Asumsikan fungsi rekursi tersebut akan menghasilkan suatu recursion tree, yang dengan level tree dimulai dari level 0, dan level paling terakhir pada tree tersebut adalah k. Tentukanlah height dan jumlah level dari tree tersebut dalam n!
- Gambarkanlah tree yang dihasilkan, hingga pada level 4. Cantumkan level, term dominan pada tiap level, serta cost tiap level! Gunakan contoh template di bawah untuk menggambarkan tree!

[Hint: Gunakan juga contoh gambar ini untuk menjawab poin b]

**Fungsi Rekursi:**  
 $T(n) = 2T(n/2) + cn; n \geq 1$   
 $T(n) = c; n = 0$



- Untuk jumlah node besar, apakah tree tersebut penuh pada tiap level nya? Tentukanlah pada level berapa tree tersebut tidak penuh lagi (dalam n)!
- Tentukanlah notasi asimtotik dari fungsi rekusi tersebut!

**3. Soal 3 (6 + 6 + 6 + 6 = 24 Poin)**

Tentukanlah notasi asimtotik dari fungsi berikut dengan metode yang ditentukan! Berikan justifikasi mengapa metode non-extended master theorem tidak berlaku pada bagian (a) dan (b).

- a.  $T(n) = 4T\left(\frac{n}{2}\right) + n^2 \lg(n)$  [substitution]
- b.  $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\lg(n)}$  [master theorem extended version]
- c.  $T(n) = 4T\left(\frac{n}{3}\right) + n$  [master theorem non-extended version]
- d.  $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$  [master theorem non-extended version]

Extended Master Theorem:

**THEOREM (Extension of Master Theorem)** If  $a, b, E \stackrel{\text{def}}{=} \log_b(a)$ , and  $f(n)$  are as in the Master Theorem, the recurrence

$$T(n) = aT(n/b) + f(n), \quad T(1) = d,$$

has solution as follows:

- 1') If  $f(n) = O(n^E (\log_b n)^\alpha)$  with  $\alpha < -1$ , then  $T(n) = \Theta(n^E)$ .
- 2') If  $f(n) = \Theta(n^E (\log_b n)^{-1})$ , then  $T(n) = \Theta(n^E \log_b \log_b(n))$ .
- 3') If  $f(n) = \Theta(n^E (\log_b n)^\alpha)$  with  $\alpha > -1$ , then  $T(n) = \Theta(n^E (\log_b n)^{\alpha+1})$ .

**4. Soal 4 (5 + 10 = 15 poin)**

Diberikan pseudocode algoritma convolution yang berfungsi untuk menghitung discrete convolution dari 2 buah vector dimensi sebagai berikut:

```
discrete_convolution(f, g):  
  
    nf = length[f]  
  
    ng = length[g]  
  
    nres = nf + ng - 1  
  
    let res[0...nres - 1] be new arrays  
  
    fp = f + [0] * (nres - nf)
```

```

gp = g + [0] * (nres - ng)

for n in range(nres):

    for m in range(n+1):

        res[n] += fp[m] * gp[n-m]

return result

```

- a) Hitunglah keluaran dari fungsi discrete\_convolution dengan input  $f = [0, 2.5, 5.0, 7.5, 10.0]$  dan  $g = [2, 4, 6, 8, 10]$ . Jelaskan tahapan-tahapannya secara detail!
- b) Hitunglah running time total  $T(n)$  algoritma tersebut dengan menghitung jumlah dieksekusinya tiap baris dalam pseudocode dan tentukan kompleksitas waktu algoritma tersebut dalam notasi big  $\Theta$ .