

# Desain dan Analisis Algoritma

Semester Ganjil 2023/2024

## PR 3 - Order Statistics, Dynamic Programming, Greedy Algorithm

**Deadline:** Senin, 20 November 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 PR3 NPM Nama.pdf.  
Contoh: PR3 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.**

**Soal 1 - Order Statistics** (5 + 10 + 15 = 30 poin)

Diberikan algoritma berikut yang menerima input 2 buah array terurut  $X$  dan  $Y$  yang masing-masing panjangnya  $n$ .

TWO-ARRAY-XXX( $X, Y, n$ )

```
1:  $x \leftarrow \text{FIND-XXX}(X, Y, n, 1, n)$ 
2: if  $x = \text{NOT-FOUND}$  then
3:    $x \leftarrow \text{FIND-XXX}(Y, X, n, 1, n)$ 
4: return  $x$ 
```

FIND-XXX( $A, B, n, low, high$ )

```
1: if  $low > high$  then
2:   return NOT-FOUND
3: else
4:    $k \leftarrow \lfloor (low + high)/2 \rfloor$ 
5:   if  $k = n$  and  $A[n] \leq B[1]$  then
6:     return  $A[n]$ 
7:   else if  $k < n$  and  $B[n - k] \leq A[k] \leq B[n - k + 1]$  then
8:     return  $A[k]$ 
9:   else if  $A[k] > B[n - k + 1]$  then
10:    return FIND-XXX( $A, B, n, low, k - 1$ )
11:  else
12:    return FIND-XXX( $A, B, n, k + 1, high$ )
```

- (a) Tentukan nilai  $x$  yang dikembalikan oleh fungsi TWO-ARRAY-XXX jika diberikan input array  $X = [16, 19, 22, 25, 30, 39]$  dan  $Y = [11, 20, 27, 28, 36, 40]$ . Apa yang dilakukan oleh algoritma TWO-ARRAY-XXX tersebut?
- (b) Tentukan kompleksitas waktu dari algoritma TWO-ARRAY-XXX dalam notasi Big- $\mathcal{O}$  untuk kasus best-case dan worst-case.
- (c) Jelaskan apa yang dilakukan algoritma tersebut pada setiap pemilihan cabang **if-then-else** dalam fungsi FIND-XXX.

## Soal 2 - Dynamic Programming (10 + 10 + 15 = 35 poin)

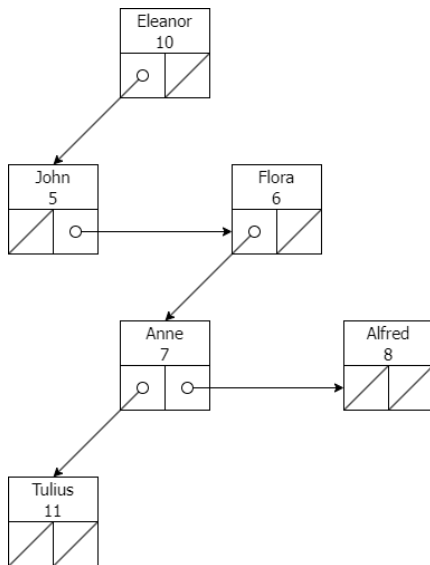
### Problem: Pesta Kerajaan

Eleanor adalah seorang Ratu di Kerajaan Kencana. Suatu hari, ia sedang mengorganisir pesta besar di istananya. Namun, Kerajaan Kencana memiliki struktur hierarki yang unik, di mana kedudukan bangsawan-bangsawannya diatur dalam struktur pohon, dengan Ratu Eleanor berada di akar.

Penasihat Kerajaan Kencana telah memberi nilai kepada setiap bangsawan dengan ‘nilai keramahan’, yang merupakan suatu angka riil. Selain itu, untuk memastikan harmoni selama pesta, Ratu Eleanor juga memutuskan bahwa tidak ada seorang bangsawan beserta bawahan langsungnya (pada pohon bangsawan) yang menghadiri pesta secara bersamaan.

Ratu Eleanor memberi Anda pohon yang menggambarkan hierarki bangsawan di Kerajaan Kencana, menggunakan representasi *left-child, right-sibling*. Setiap node pohon berisi, selain pointer, nama bangsawan dan nilai keramahan bangsawan tersebut. Tugas Anda adalah membuat daftar undangan yang memaksimalkan jumlah total nilai keramahan dari para peserta pesta.

Contoh:



Di sini, Ratu Eleanor memiliki bawahan langsung 2 orang bangsawan, yaitu John dan Flora. Flora sendiri juga memiliki bawahan langsung 2 orang bangsawan, yaitu Anne dan Alfred. Anne punya seorang bawahan langsung yaitu Tullius. Maka yang perlu diundang supaya jumlah nilai keramahannya maksimal adalah Eleanor, Alfred, dan Tullius dengan total nilai keramahan 29.

(a) Definisikan substruktur yang optimal dari permasalahan tersebut.

Untuk setiap node  $x$  pada pohon, kita perlu menghitung dua nilai:

- Nilai maksimal total keramahan yang diperoleh dari sub-tree  $x$  apabila  $x$  diundang  
 $x.invited = \dots$
- Nilai maksimal total keramahan yang diperoleh dari sub-tree  $x$  apabila  $x$  tidak diundang  
 $x.not\_invited = \dots$

(b) Buatlah algoritma  $\text{MAXIMIZEKERAMAHAN}(x)$  yang menerima input node  $x$  dan menghitung dua nilai, yaitu  $x.invited$  dan  $x.not\_invited$  dan menyimpannya. Tentukan kompleksitas waktunya dalam notasi big- $\Theta$ .

(c) Buatlah algoritma  $\text{OUTPUTUNDANGAN}(P)$  yang mencetak daftar tamu undangan dari tree dengan root  $P$ . Gunakan nilai  $invited$  dan  $not\_invited$  yang telah dihitung sebelumnya dengan algoritma  $\text{MAXIMIZEKERAMAHAN}$ . Tentukan kompleksitas waktunya dalam notasi big- $\Theta$ .

### Soal 3 - Greedy Algorithm (10 + 15 + 10 = 35 poin)

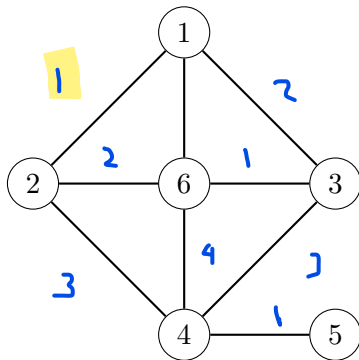
#### Problem: Graph Edge-Coloring

Diberikan algoritma EDGE-COLORING berikut yang menerima input suatu graf yang direpresentasikan dengan array *edgesList* berisikan semua edge-nya. Algoritma tersebut mengembalikan array *colors* yang berisikan pewarnaan setiap edge dalam graf tersebut, serta memastikan bahwa setiap edge yang bersisian dengan vertex yang sama tidak memiliki warna identik.

EDGE-COLORING(*edgesList*)

```
1:  $e \leftarrow \text{length of } edgesList$ 
2:  $colors \leftarrow \text{list of size } e \text{ initialized with zeros}$ 
3:  $i \leftarrow 0$ 
4:  $color \leftarrow 1$ 
5: while  $i < e$  do
6:    $colors[i] \leftarrow color$ 
7:    $flag \leftarrow \text{false}$ 
8:   for  $j$  from 0 to  $e - 1$  do
9:     if  $j \neq i$  then
10:       $commonVertices \leftarrow \text{intersection of } edgesList[i] \text{ and } edgesList[j]$ 
11:      if  $commonVertices$  is not empty and  $colors[i] = colors[j]$  then
12:         $color \leftarrow color + 1$ 
13:         $flag \leftarrow \text{true}$ 
14:        break
15:   if not  $flag$  then
16:      $color \leftarrow 1$ 
17:      $i \leftarrow i + 1$ 
18: return  $colors$ 
```

(a) Diberikan graph sebagai berikut.



Tentukan hasil pewarnaan setiap edge-nya apabila algoritma EDGE-COLORING dijalankan dengan input  $edgeList = [(1, 2), (1, 6), (1, 3), (2, 6), (6, 4), (3, 6), (3, 4), (2, 4), (4, 5)]$

- (b) Jelaskan langkah-langkah yang dilakukan oleh algoritma EDGE-COLORING tersebut. Mengapa algoritma tersebut bersifat greedy?
- (c) Apakah algoritma tersebut selalu memberikan hasil pewarnaan dengan banyak warna minimal? Jelaskan jika ya, dan berikan counterexample jika tidak.  
(Hint: Apakah urutan pewarnaan edge berpengaruh terhadap hasil akhir pewarnaan?)