

# Desain dan Analisis Algoritma

Semester Genap 2023/2024

## PR 3 – Sorting in Linear Time, Randomized Algorithm, Dynamic Programming

Deadline: **Senin, 29 April 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 PR1 NPM Nama.pdf.  
Contoh: PR1 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 dije- laskan. 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 (30 poin)

Diberikan *pseudocode* beserta contoh ilustrasi untuk simulasi *sorting* dari *counting sort*:

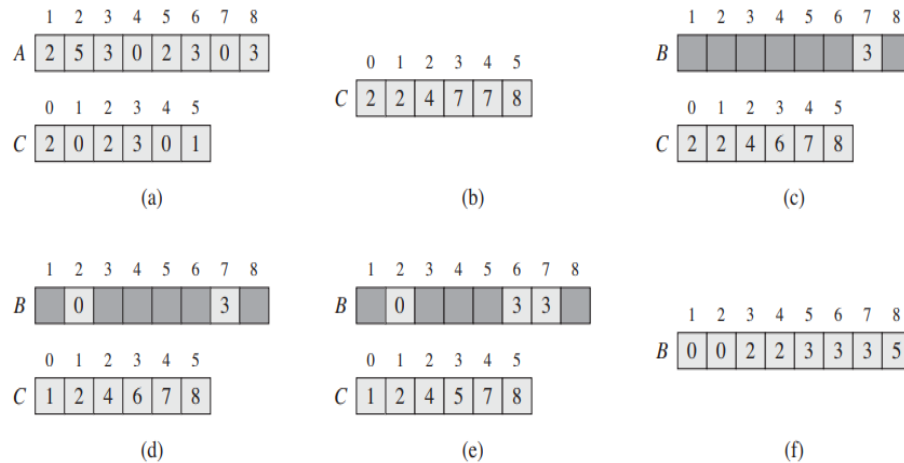
COUNTING-SORT( $A, B, k$ )

```

1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 

```

Gambar 1: Pseudocode



Gambar 2: Simulasi

- Modifikasilah pseudocode tersebut agar dapat melakukan *sorting* secara *descending* dan *stable*!
- Dengan ilustrasi simulasi dan algoritma pada bagian (a), lakukan proses dan tunjukkan simulasinya untuk mengurutkan array  $[3, 2, 9, 0, 6, 7]$  ditambah dengan seluruh digit NPM Anda secara *descending*!
- Diberikan simulasi untuk *radix sort* untuk 7 buah angka secara *ascending* sebagai berikut:

329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

Urutkanlah 10 buah kata dengan 5 huruf berikut secara *alphanumeric (ascending)* dengan *radix sort*, sesuai dengan ilustrasi di atas: MAKAN, MINUM, BOTOL, KARTU, GELAS, PINTU, DAPUR, HANTU, PAGAR, BUNGA

2. Soal 2 (15 poin)

**Quicksort**

Berikut adalah algoritma *quicksort* menggunakan partisi Lomuto:

QUICKSORT(A, p, r)

```
1  if p < r
2    q = PARTITION(A, p, r)
3    QUICKSORT(A, p, q - 1)
4    QUICKSORT(A, q + 1, r)
```

PARTITION(A, p, r)

```
1  x = A[r]
2  i = p - 1
3  for j = p to r - 1
4    if A[j] <= x
5      i = i + 1
6      exchange A[i] with A[j]
7  exchange A[i + 1] with A[r]
8  return i + 1
```

- Demonstrasikan *quicksort* untuk masukan  $A = [3, 6, 1, 9, 4, 2, 8, 1, 7]$ ,  $p = 1$ , dan  $r = 9$  (asumsi *array one-indexing*) hingga terurut dengan menuliskan *array* beserta *pointer*  $i$  dan  $j$  setelah setiap *exchange* dalam PARTITION. Apakah keluaran *array* yang Anda dapatkan?
- Apakah *quicksort* bersifat stabil? Bagian mana dalam *pseudocode quicksort* yang menjamin stabilitas atau berpotensi menyebabkan ketidakstabilan?
- Apakah perbedaan antara *randomized quicksort* dan *quicksort* biasa dari segi kompleksitas waktu dalam berbagai kasus? Berikan contoh masukan *array* yang menunjukkan perbedaan tersebut beserta alasannya.

3. Soal 3 (25 poin)

**Randomized Algorithm**

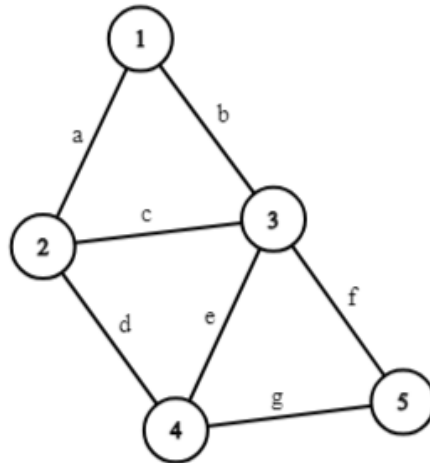
Berikut adalah sebuah algoritma yang bertujuan mencari *min-cut* (jumlah terkecil *edge* yang dihapus untuk memecah sebuah graf menjadi dua subgraf) dalam graf tak berarah dan tak berbobot dengan himpunan *vertex*  $V$  dan himpunan *edge*  $E$ :

FIND-MIN-CUT( $V, E$ )

```

1  copy  $V$  into  $V'$ 
2  copy  $E$  into  $E'$ 
3  while  $|V'| > 2$ 
4    pick random edge  $(u, v) \in E'$ 
5    add new vertex  $uv$  to  $V'$ 
6    for all edges  $(x, y) \in E'$ 
7      if  $x == u$  or  $x == v$ 
8         $x = uv$ 
9      if  $y == u$  or  $y == v$ 
10        $y = uv$ 
11     if  $x == y$ 
12       remove  $(x, y)$  from  $E'$ 
13   remove  $u$  from  $V'$ 
14   remove  $v$  from  $V'$ 
15 return  $E'$ 

```



- Demonstrasikan algoritma tersebut untuk masukan graf di atas dan urutan *edge* yang dihapus [e, g, a] dengan menggambarkan graf setelah setiap iterasi *while loop*. Berapa *edge cut* yang Anda dapatkan?
- Apakah algoritma tersebut termasuk *randomized algorithm* Monte Carlo atau Las Vegas? Berikan alasan untuk mendukung jawaban Anda (Petunjuk: apakah jawaban pada subnomor a merupakan *min-cut*?).
- Jika Anda menjawab Monte Carlo pada subnomor b:** Berapakah peluang ditemukannya *min-cut* setelah algoritma dijalankan sekali? Berapakah peluangnya setelah dijalankan sebanyak jumlah *vertex*?  
**Jika Anda menjawab Las Vegas pada subnomor b:** Hitunglah kompleksitas waktu algoritma tersebut dalam notasi big O. Sertakan asumsi jika perlu.

4. Soal 4 (30 poin)

**Dynamic Programming**

Andi petualang sedang merencanakan perjalanan wisata di sebuah pulau yang penuh dengan keajaiban alam. Di pulau tersebut, terdapat serangkaian tempat wisata yang saling terhubung, dinomori dari 1 sampai N. Setiap tempat wisata memiliki sebuah harta karun dengan nilai tertentu dan beratnya yang berbeda-beda. Namun, Andi hanya dapat membawa harta karun dengan total berat tertentu dalam kopernya. Selain itu, Andi harus mengikuti aturan yang unik yaitu begitu dia mengunjungi suatu tempat wisata (i), maka pintu yang menghubungkannya dengan tempat wisata berikutnya (i+1) akan tertutup, dan sebuah lorong rahasia menuju tempat wisata setelahnya (i+2) akan terbuka. Selain itu, Andi tidak bisa kembali ke tempat wisata sebelumnya (i-1). Dengan kata lain, dia tidak dapat mengunjungi dua tempat wisata yang bersebelahan.

Andi ingin merencanakan perjalanan yang memaksimalkan total nilai harta karun yang akan dia ambil, dengan memperhatikan batasan berat yang dimilikinya.

- Tentukan solusi rekursif dari permasalahan di atas termasuk parameter dari fungsi rekursif tersebut.
- Rancanglah solusi bottom-up DP dari permasalahan di atas dan implementasikan dalam bahasa pemrograman yang kalian suka.
- Tentukan kompleksitas running time dari solusi tersebut.
- Terapkan algoritma tersebut untuk mencari solusi optimal untuk membantu Andi petualang yang akan pergi ke sebuah pulau dengan 8 tempat wisata, dengan masing-masing tempat wisata memiliki nilai harta karun secara berurutan 10, 5, 2, 3, 8, 4, 7, dan 6, serta beratnya secara berurutan 3, 2, 4, 1, 6, 5, 7, dan 4, sedangkan berat maksimum koper Andi adalah 10.

5. Soal Bonus (30 poin)

**Word Segmentation Problem**

Di dunia ini, ada beberapa bahasa dimana beberapa kata penyusun digabung menjadi satu sehingga tidak terlihat batas antar kata. Beberapa bahasa tersebut adalah Bahasa Mandarin; dan beberapa bahasa di Eropa yang terkadang menggabung beberapa kata menjadi satu.

Diberikan sebuah string yang terdiri dari  $n$  huruf, temukan solusi efisien untuk melakukan segmentasi string tersebut menjadi kata-kata sehingga memberikan **skor yang tertinggi**.

Selain string tersebut, program juga diberikan input berupa dictionary yang berisi nilai skor beberapa sub-string. Contoh:

Sub-string	Skor
mogen	4
enz	1
gen	2
sla	2
pen	2
slapen	5
ze	1
en	1

Contoh: "mogenzeslapen"

Score(mo, gen, ze, sla, pen) = 7

Score(mogen, ze, slapen) = 10 (optimal)