

*Berdoalah terlebih dahulu agar Anda berhasil dalam ujian ini!*

### **Brute Force + Divide and Conquer**

1. Diberikan sebuah larik yang berisi elemen biner (0 atau 1). Elemen-elemen larik sudah **terurut** menaik (dari kecil ke besar). Kita akan menghitung jumlah bit 1 di dalam larik tersebut. Contoh:  $A = [0, 0, 0, 1, 1, 1, 1, 1]$ , jumlah bit 1 adalah 5.
  - (a) **(Nilai: 10)** Jika diselesaikan dengan algoritma brute force, jelaskan caranya atau langkah-langkahnya, lalu tentukan kompleksitas waktu asimptotiknya dalam notasi O besar.
  - (b) **(Nilai: 13)** Jika diselesaikan dengan algoritma divide and conquer, jelaskan langkah-langkahnya (tuliskan bagian basis dan rekurensya). Tentukan kompleksitas waktunya dalam bentuk rekursif, lalu selesaikan untuk menentukan kompleksitas waktu asimptotiknya dalam notasi O besar. Ilustrasikan jawaban anda dengan contoh  $A = [0, 0, 0, 1, 1, 1, 1, 1]$

#### **Jawaban:**

- (a) Traversal larik sampai ketemu elemen 1 pertama, misalkan pada posisi k. Karena larik sudah terurut, maka hitung jumlah 1 dari posisi tersebut sampai posisi terakhir dengan rumus  $\text{count} = (n - k) + 1$ . Kompleksitas waktu algoritamanya, dihitung dari jumlah operasi perbandingan elemen, adalah  $O(n)$
- (b) Algoritma:

#### **HitungBit1(A, n, count)**

- (i) Basis: Jika ukuran larik = 0, maka jumlah bit 1 sama dengan 0
- (ii) Rekurens: Jika ukuran larik > 0, maka
  - Bagidua larik pada posisi pertengahan, A1 dan A2, masing-masing larik berukuran  $n/2$
  - **jika** pada upalarik kiri, A1, elemen terakhirnya 0, maka dipastikan seluruh elemen larik pada bagian tersebut adalah 0 (karena larik sudah terurut), jadi  $\text{count1} = 0$ .  
**else** HitungBit1(A1,  $n/2$ , count1)
  - **jika** elemen pertama pada upalarik kanan, A2, adalah 1, maka dipastikan seluruh elemen pada bagian larik tersebut adalah 1. Jadi,  $\text{count2} = \text{ujung kanan} - \text{ujung kiri} + 1$ .  
**else** HitungBit1(A2,  $n/2$ , count2)
  - $\text{count} = \text{count1} + \text{count2}$  { tahap combine }

Kompleksitas waktu algoritma, dihitung dari jumlah operasi perbandingan elemen, adalah:

$$T(n) = a, \text{ jika } n = 0$$
$$= T(n/2) + 2, \text{ jika } n > 0$$

Menurut teorema master,  $a = 2$ ,  $b = 2$ ,  $d = 0$ , sehingga  $a = b^d$  (case 2), jadi  $T(n) = O(n^0 \log n) = O(\log n)$

2. **(Nilai : 3 + 3 + 3 + 3)** Dengan menggunakan teorema master, tentukan notasi Big-Oh untuk kompleksitas waktu berikut:

- a.  $T(n) = 8T(n/2) + 1000n^2$
- b.  $T(n) = 5T(n/4) + 5n$
- c.  $T(n) = 25T(n/5) + n^2$
- d.  $T(n) = 2T(n/4) + \sqrt{n} + 42$ .

**Jawaban:**

- a.  $O(n^3)$
- b.  $O(n^4 \log 5)$
- c.  $O(n^2 \log n)$
- d.  $O(\sqrt{n} \log n)$

### ***Exhaustive Search & Greedy***

3. Seorang mahasiswa IF tingkat 4 mengalokasikan waktu  $n$  jam untuk mengerjakan proyek programming. Terdapat  $m$  proyek ( $p_1..p_m$ ) yang mungkin diterima dengan mempertimbangkan estimasi waktu kerja ( $t_1..t_m$ ) dalam jam, dan honor yang akan diterima ( $s_1..s_m$ ) dalam ratusan ribu. Bantulah mahasiswa ini memilih proyek yang akan dikerjakan untuk memaksimalkan honor yang akan diterima. Pertanyaan a-d dijawab dengan asumsi proyek dikerjakan secara utuh (diambil atau tidak diambil).
  - a) **(Nilai: 3+3+3)** Jika diselesaikan dengan exhaustive search, jelaskanlah langkah-langkah yang dilakukan di dalam exhaustive search. Lengkapilah penjelasan dengan jumlah kandidat solusi yang akan dievaluasi, dan tentukan kompleksitas waktu asimptotiknya dalam notasi  $O$  besar.
  - b) **(Nilai: 8)** Jika diselesaikan dengan greedy, definisikanlah elemen-elemen dari algoritma greedy untuk persoalan ini dengan menyatakan secara eksplisit anggotanya jika elemen greedy berupa himpunan dan definisi fungsi (domain  $\rightarrow$  fungsi) dan deskripsinya jika elemen greedy berupa fungsi. Strategi greedy disebutkan secara eksplisit pada fungsi seleksi.
  - c) **(Nilai: 5)** Ilustrasikan jawaban anda dengan strategi greedy jika waktu yang dialokasikan proyek sebanyak 60 jam, terdapat 4 proyek A..D dengan estimasi waktu  $t_i$  dan honor  $s_i$  sebagai berikut.  
 Proyek A:  $t=40, s=2800$   
 Proyek B:  $t=10, s=1000$   
 Proyek C:  $t=20, s=1200$   
 Proyek D:  $t=24, s=1200$   
 Tentukanlah himpunan solusi yang dihasilkan, dan total honor yang akan diterima.
  - d) **(Nilai: 3)** Apakah solusi yang diberikan di bagian c sudah optimal? Jelaskanlah pendapat Anda.
  - e) **(Nilai: 5)** Misalkan ada kesempatan untuk mengerjakan proyek secara part time, berikanlah ilustrasi jawaban Anda dengan strategi greedy untuk persoalan c.

**Jawaban:**

- a) Langkah-langkah dalam exhaustive search (nilai 3):
  - i) enumerasi kandidat solusi dengan sistematis. Jumlah kandidat solusi:  $2^n$  (Nilai 3)
  - ii) evaluasi setiap kandidat solusi, simpan solusi terbaik.
  - iii) umumkan solusi terbaik di akhir pencarian.  
Kompleksitas waktu  $O(n \cdot 2^n)$ . (Nilai 3)
- b) **Himpunan kandidat (C):** berisi  $m$  proyek  $p_1..p_m$   
**Himpunan solusi (S):** berisi proyek  $p_i$  yang sudah dipilih  
**Fungsi solusi:** himpunan solusi  $S \rightarrow$  boolean, menentukan apakah  $S$  sudah merupakan solusi lengkap.  
**Fungsi seleksi:** himpunan kandidat  $C \rightarrow$  kandidat, menentukan proyek yang memiliki densitas tertinggi dari  $C$ . Pada setiap langkah, *knapsack* diisi dengan proyek yang mempunyai densitas ( $p_i / w_i$ ) terbesar. (Nilai 3)  
**Fungsi kelayakan:** himpunan  $S \cup$  kandidat  $\rightarrow$  boolean, true jika proyek yg dipilih menjadi bagian dari  $S$  maka waktu yang dibutuhkan tidak melebihi alokasi waktu  $n$  jam.

**Fungsi obyektif:** memaksimalkan honor yang diterima.

c)

Proyek	t	s	s/t	Dipilih
A	40	2800	70	1
B	10	1000	100	1
C	20	1200	60	0
D	24	1200	50	0

Solusi:  $\langle 1, 1, 0, 0 \rangle$

Total honor=3800 ratus ribu

d) Tidak, solusi optimal  $\langle 1, 0, 1, 0 \rangle$  dengan total honor 4000 ratus ribu.

e)

Proyek	t	s	s/t	Dipilih
A	40	2800	70	1
B	10	1000	100	1
C	20	1200	60	0.5
D	24	1200	50	0

Solusi:  $\langle 1, 1, 0.5, 0 \rangle$

Total honor=4400 ratus ribu

4. **(Nilai: 5)** Sebutkanlah 2 persoalan yang dipelajari di kuliah yang tidak menjamin dihasilkannya solusi optimal oleh strategi greedy. Jelaskanlah untuk setiap persoalan tersebut fungsi obyektif, strategi greedy, dan contoh instansiasi kasus yang diilustrasikan tidak menghasilkan solusi optimal.

**Jawaban:**

1. Persoalan penukaran uang, dgn penjelasan
2. Persoalan 1/0 knapsack, dgn penjelasan

### *Decrease and Conquer*

5. **(Nilai 7)** Terdapat beberapa algoritma untuk mencari pembagi bilangan terbesar (Greatest Common Divisor/ GCD) yang sudah anda pelajari. Jelaskan salah satu algoritma untuk mencari GCD dari dua buah bilangan bulat positif, yang menggunakan pendekatan Decrease and Conquer. Jelaskan langkah-langkah nya (bukan pseudo code), dan terapkan pada contoh mencari GCD(10,15) dan GCD(20,12).

**Jawaban:**

Dengan algoritma Euclidean yang sudah dipelajari di kuliah Matdis. Misal parameter pertama m, parameter kedua n:

1. jika  $n = 0$ , kembalikan m sebagai hasil GCD (basis), else
2. m dicari sisa pembagian (mod, misal disimpan di r) dengan n
3. panggil kembali (rekursif) GCD dengan nilai m diganti n, dan nilai n diganti r, hingga dicapai basis.

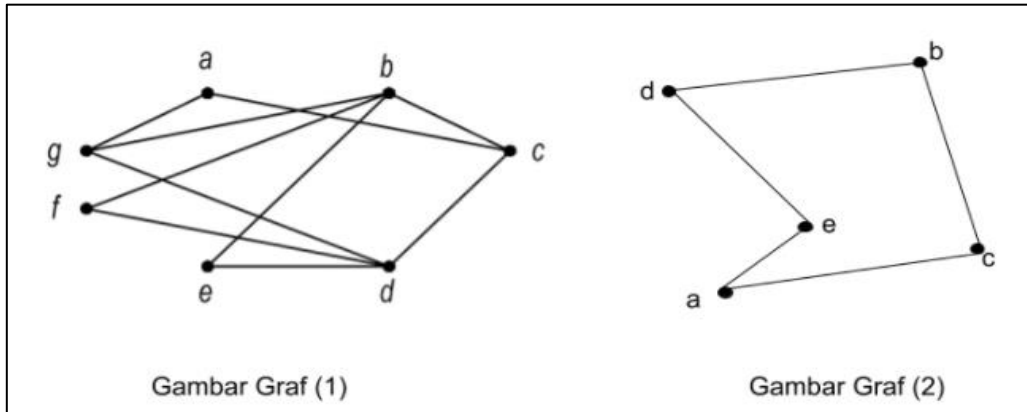
Bilangan pada parameter di setiap langkah tersebut akan terus berkurang (decrease), dan ini bagian yang di 'conquer'. Tidak ada tahap combine pada decrease and conquer.

$$\text{GCD}(10,15) = \text{GCD}(15,10) = \text{GCD}(10,5) = \text{GCD}(5,0) = 5$$

$$\text{GCD}(20,12) = \text{GCD}(12,8) = \text{GCD}(8,4) = \text{GCD}(4,0) = 4$$

### *DFS - BFS*

6. Terdapat sebuah graf terhubung sederhana tidak berarah seperti pada Gambar 1 berikut ini. Setiap simpul akan diberi warna, dan warna yang digunakan hanya warna merah dan biru. Diasumsikan sudah ada matriks ketetanggaan yang merepresentasikan graf tersebut, untuk mengetahui sebuah simpul bertetangga dengan simpul mana saja. Urutan penelusuran atau pemeriksaan simpul tetangga mengikuti urutan abjad.



Gambar 1. Graf Terhubung Sederhana Tidak Berarah

- a. **(Nilai 7)** Jika pewarnaan Graf (1) dimulai dari simpul a dengan warna biru, penelusuran simpul berikutnya mengikuti aturan DFS, dan simpul tetangganya (misal simpul x) diberi warna merah, lalu simpul tetangga dari simpul x diberi warna biru hingga akhir simpul berselang seling pemberian warnanya dengan simpul tetangga, tuliskan warna yang diberikan pada setiap simpul dan penulisan diurutkan berdasarkan urutan penelusurannya. Contoh penulisan mulai dari simpul a:

1. Simpul a - biru
2. Simpul x - merah
- ... dst

**Jawaban:**

1. **Simpul a - biru**

2. **Simpul c** (karena c lebih awal daripada g pada urutan abjad) - merah

3. **Simpul b - biru**

4. **Simpul e - merah**

5. **Simpul d - biru**

6. **Simpul f** (karena f lebih awal daripada g pada urutan abjad) - **merah** (karena tidak ada tetangga f yg belum ditelusuri, backtrack ke d, periksa tetangga d yg lain)

7. **Simpul g - merah** (ingat pemberian warna berselang seling dengan simpul tetangga)

8. Selesai (semua simpul sudah ditelusuri)

- b. **(Nilai 5)** Jika pewarnaan Graf (2) dimulai dari simpul a dengan warna biru, penelusuran simpul berikutnya mengikuti aturan BFS, dan simpul tetangganya (misal ada simpul x) diberi warna merah, lalu simpul tetangga dari simpul x diberi warna biru hingga akhir simpul berselang seling pemberian warnanya dengan simpul tetangga, tuliskan warna yang diberikan pada setiap simpul dan penulisan diurutkan berdasarkan urutan penelusurannya. Contoh penulisan mulai dari simpul a:

1. Simpul a - biru
2. Simpul x - merah
- ... dst

**Jawaban:**

1. **Simpul a - biru**

2. **Simpul c** (karena c lebih awal daripada e pada urutan abjad) - **merah**

3. **Simpul e - merah** ((ingat pemberian warna berselang seling dengan simpul tetangga, dan e adalah tetangga simpul a)

4. **Simpul b - biru** (karena simpul b adalah tetangga simpul c)

5. **Simpul d - biru** (karena simpul d adalah tetangga simpul e dari urutan proses penelusuran secara BFS)

c. **(Nilai 1)** Apakah ada simpul yang saling bertetangga memiliki warna yang sama untuk jawaban butir (a)?  
Simpul apa saja jika ada?

**Jawaban:** Tidak ada

d. **(Nilai 1)** Apakah ada simpul yang saling bertetangga memiliki warna yang sama untuk jawaban butir (b)?  
Simpul apa saja jika ada?

**Jawaban:** Ada, simpul b dan simpul d.

e. **(Nilai 2)** Apakah Graf (1) bipartite? Jelaskan jawaban dan kaitkan dengan jawaban di butir (c).

**Jawaban:** Jika semua simpul dengan warna yang sama dikelompokkan menjadi satu himpunan bagian, maka terdapat dua himpunan bagian (misal V1 dan V2), dan setiap sisi menghubungkan sebuah simpul di V1 ke simpul di V2, dan tidak ada simpul di V1 yg saling bertetangga, demikian juga simpul di V2. Sehingga Graf (1) Bipartite.

f. **(Nilai 2)** Apakah Graf (2) bipartite? Jelaskan jawaban dan kaitkan dengan jawaban di butir (d).

**Jawaban:** Jika setiap simpul dengan warna yang sama dikelompokkan menjadi satu himpunan bagian, maka terdapat simpul di salah satu himpunan bagian yang memiliki sisi dengan simpul warna yang sama, yaitu simpul b dan simpul d. Jadi Graf (2) tidak bipartite.

g. **(Nilai 5)** Apakah simpul d bisa dicapai dari simpul awal b pada Graf (1), jika penelusuran menggunakan DLS dengan batas kedalaman 3? Tuliskan urutan langkah pencariannya (urutan tetangga sesuai urutan abjad) termasuk jika ada backtrack; dan jika berhasil ditemukan, tentukan kedalaman simpul d dalam pohon pencarian tersebut.

**Jawaban:** Jika setiap simpul dengan warna yang sama dikelompokkan menjadi satu himpunan bagian, maka terdapat simpul di salah satu himpunan bagian yang memiliki sisi dengan simpul warna yang sama, yaitu simpul b dan simpul d. Jadi Graf (2) tidak bipartite.