

Ujian Tengah Semester **IF2211 Strategi Algoritma**

Senin, 6 April 2026

Waktu: 100 menit

Dosen: Rinaldi M (K1), Rila Mandala (K2), Fazat Nur Azizah & Tricya Widagdo (K3)

---

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

1. **(Brute Force + Greedy, Nilai = 15)** Seorang petugas kebersihan harus memroses  $n$  tugas. Setiap tugas memiliki durasi ( $a$ ) dan waktu tenggat (*deadline*) ( $d$ ) dan petugas harus memroses setiap tugas berturut-turut (satu tugas harus selesai terlebih dahulu sebelum memulai tugas berikutnya). Bonus untuk suatu tugas adalah selisih  $d - f$  dengan  $d$  adalah tenggat untuk tugas yang bersangkutan dan  $f$  adalah waktu penyelesaian tugas tersebut. Waktu dimulai dari 0 dan semua tugas harus diproses walaupun menghasilkan bonus negatif.

(a) Buatlah algoritma *brute force* (bukan *pseudo-code*) untuk menentukan jumlah maksimum bonus yang dapat diterima dari pengerjaan semua tugas dengan masukan:

$n$  = banyaknya tugas

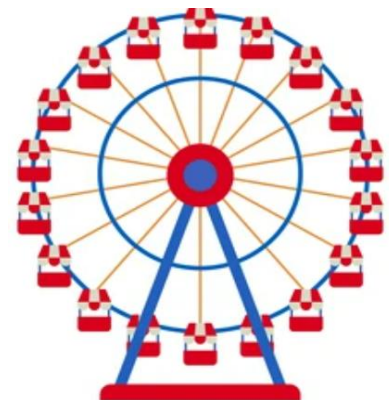
Daftar tugas =  $\{(a_1, d_1), (a_2, d_2), \dots, (a_n, d_n)\}$ , dengan  $a_i$  adalah durasi penyelesaian tugas  $i$  dan  $d_i$  adalah waktu tenggat tugas  $i$ .

Berikan ilustrasi algoritma Anda dengan menggunakan kasus berikut:  $n = 6$ , daftar tugas =  $\{(6,10), (5,12), (4,19), (5,20), (4,17), (6,20)\}$  (Keterangan: (6,10) artinya durasi penyelesaian 6 jam dan harus selesai sebelum jam 10)

Tentukan perkiraan kompleksitas algoritmanya dalam notasi big-O

(b) Ulangi soal (a) di atas tetapi dengan pendekatan algoritma *greedy*. Berikan ilustrasi algoritma Anda dengan menggunakan kasus berikut:  $n = 6$ , daftar tugas =  $\{(6,10), (5,12), (4,19), (5,20), (4,17), (6,20)\}$

2. **(Greedy, Nilai = 15)** Setiap tahun Pak Budi mengajak siswa dari kelasnya untuk mengunjungi suatu taman bermain. Salah satu permainan yang sangat disukai adalah kincir ria (*ferris wheel*). Setiap gondola di kincir ria bisa dinaiki maksimum 4 orang dan total berat penumpang untuk tiap gondola tidak boleh melebihi suatu nilai  $x$ . Biaya karcis untuk sebuah gondola sangat mahal sehingga Pak Budi ingin sesedikit mungkin menyewa gondola untuk siswa-siswanya. Buatlah algoritma dengan pendekatan *greedy* untuk menghitung jumlah minimum gondola yang dibutuhkan untuk mengangkut semua siswa, jika diberikan: banyaknya siswa  $n$ , berat maksimum per gondola  $x$ , dan daftar berat siswa  $w_1, w_2, \dots, w_n$ .



Jelaskan ilustrasi algoritma Anda untuk kasus berikut:  $n = 10$ ;  $x = 55$ . Daftar berat badan siswa =  $[15, 20, 10, 10, 17, 18, 15, 25, 14, 16]$ . Jelaskan apakah algoritma akan memberikan hasil yang optimal? Buktikan!

3. **(Divide and Conquer, Nilai = 12)** Diketahui sebuah persoalan *rahasia*. Persoalan *rahasia* tersebut diselesaikan dengan menggunakan algoritma *divide and conquer* sebagai berikut

**Procedure** *rahasia* (input  $n$  : integer)

**Algoritma**

if  $n = 1$  solve()

else

*rahasia*( $n/2$ )

*rahasia*( $n/4$ )

*rahasia*( $n/4$ )

*berubah*()

end if

Jika kompleksitas waktu dari procedure *solve()* adalah 1 dan kompleksitas dari procedure *berubah()* adalah  $n$ , jawablah pertanyaan di bawah ini :

- Tuliskan kompleksitas waktunya  $T(n)$  secara rekuren.
- Jika tiap simpul dalam pohon rekurensi tersebut adalah *cost* untuk menghitung procedure *berubah()* nya saja, Gambarkan pohon rekurensinya dengan *root* adalah persoalan dengan ukuran  $n$  (sehingga *cost* di level *root* adalah  $n$ ) dan anak dari setiap simpul adalah ekspansi dari rekurensinya. Gambarkan pohonnya hanya sebanyak 3 level saja (level *root*, level *root*+1 dan level *root*+2).
- Jika tiap simpul dalam pohon rekurensi tersebut adalah *cost* untuk menghitung procedure *berubah()* nya saja, hitunglah *cost* nyatanya untuk di level *root*, level *root*+1, dan level *root*+2. (kalian harus menulis dengan penurunannya, bukan hanya jawaban akhirnya saja).
- Hitunglah kompleksitas  $T(n)$  akhirnya setelah penjabaran rekurensi nya. (kalian harus menulis dengan penurunannya, bukan hanya jawaban akhirnya saja).

4. **(Decrease and Conquer, Nilai = 13)** Diketahui sebuah program misterius yang menyelesaikan persoalan misterius dengan menggunakan algoritma *decrease and conquer* sbb :

```

procedure Misterius (input  $A : \text{LarikInteger}, i, j : \text{integer}; K : \text{integer};$  output  $id : \text{integer}$ )
Kamus
   $p : \text{integer}$ 

Algoritma:
  if  $i > j$  then
     $id \leftarrow -1$ 
  else
     $p \leftarrow i + (j - i) * (K - A(i)) / (A(j) - A(i))$ 
    if  $A(p) = K$  then
       $id \leftarrow p$ 
    else
      if  $A(p) > K$  then
        Misterius( $A, i, p - 1, K, id$ )
      else
        Misterius( $A, p + 1, j, K, id$ )
      endif
    endif
  endif

```

- Hitunglah *time complexity* dalam notasi Big-O dari procedure *Misterius* di atas untuk kasus terbaik.
- Hitunglah *time complexity* dalam notasi Big-O dari procedure *Misterius* di atas untuk kasus terburuk.
- Hitunglah *time complexity* dalam notasi Big-O dari procedure *Misterius* di atas untuk kasus rata-rata.

5. **(Decrease and Conquer, Nilai = 15)** Sebuah kode\_kuliah mungkin memiliki nol atau lebih prasyarat kuliah (*prerequisite*). Kode\_kuliah bisa diambil pada suatu semester jika semua prasyaratnya sudah pernah diambil pada semester sebelumnya (tidak harus 1 semester sebelumnya, bisa saja 2 semester sebelumnya). Asumsi semua kuliah bisa diambil pada sembarang semester, baik semester ganjil maupun semester genap. Sebagai contoh, terdapat 5 kuliah: C1, C2, C3, C4, dan C5, yang harus diambil seorang mahasiswa dengan daftar *prerequisite* sebagai berikut:

- C1, Prerequisite: C3.
- C2, Prerequisite: C1, C4.
- C3, Prerequisite: tidak ada
- C4, Prerequisite: C1, C3.
- C5, Prerequisite: C2, C4..

- Gambarkan graf *Directed Acyclic Graph* (DAG) yang merepresentasikan kelima kuliah di atas
- Lakukan *topological sort* untuk menentukan kuliah apa saja yang diambil pada setiap semester (semester 1, 2, 3, 4, dst). Perlihatkan proses *topological sort* berdasarkan graf dari jawaban (a) di atas

6. **(Divide and Conquer, Nilai = 15)** Diberikan sebuah matriks integer ukuran  $n \times n$ , asumsikan  $n$  merupakan perpangkatan dua. Kita akan menghitung selisih nilai maksimum dan nilai minimum matriks dengan algoritma divide and conquer
- Tuliskan garis besar algoritmanya (boleh dalam *pseudo-code*)
  - Tuliskan kompleksitas waktunya,  $T(n)$ , dalam bentuk relasi rekurens
  - Hitung notasi Big-Oh dengan menggunakan Teorema Master
7. **(BFS, DFS, dan DLS, Nilai = 15)** Berikut ini adalah senarai ketetanggaan yang memuat informasi graf berarah.  
*Elemen  $x : y, z$  menyatakan terdapat sisi dari simpul  $x$  ke simpul  $y$  dan simpul  $z$ .*
- { A : B, C; B : E; C : D, E; D : F, G; E : A; F : H; G : I, J; H : G, I; I : J }
- Gambarkan graf berarah sesuai informasi pada senarai ketetanggaan tersebut.
  - Jika dari simpul A ingin menuju simpul G, tentukan **jalur penelusuran/pencarian** menggunakan **masing-masing algoritma berikut ini** dan **tentukan jalur hasil** pencarian untuk masing-masing algoritma. Perhatikan, jika terdapat lebih dari satu simpul tetangga, pemilihan simpul sesuai urutan abjad.
    - BFS
    - DFS
    - DLS dengan batas kedalaman 4