

Solusi UAS IF1220 Matematika Diskrit (3 SKS)  
Dosen: Rinaldi M, Arrival Dwi Sentosa  
Rabu, 11 Juni 2025  
Waktu: 125 menit

---

Berdoalan terlebih dahulu sebelum mengerjakan ujian ini.

1. Diberikan dua bilangan bulat positif  $a = 120$  dan  $b = 75$ . Dengan menggunakan algoritma Euclidean, tentukan:

- (a) PBB( $a, b$ )
- (b) Nyatakan PBB( $a, b$ ) sebagai kombinasi linear dari  $a$  dan  $b$
- (c) Tentukan solusi dari persamaan kongruen  $120x \equiv 75 \pmod{91}$

(Nilai = 12)

**Jawaban:**

(a) Mencari PBB(120, 75) dengan Algoritma Euclidean

$$\begin{aligned} 120 &= 1 * 75 + 45 \\ 75 &= 1 * 45 + 30 \\ 45 &= 1 * 30 + 15 \\ 30 &= 2 * 15 + 0 \\ \rightarrow \text{PBB}(120, 75) &= 15. \end{aligned}$$

(b) Menyatakan PBB(120, 75) sebagai Kombinasi Linear

$$\begin{aligned} 15 &= 45 - 1 * 30 \\ 15 &= 45 - 1 * (75 - 1 * 45) \\ 15 &= 2 * 45 - 1 * 75 \\ 15 &= 2 * (120 - 1 * 75) - 1 * 75 \\ \rightarrow 15 &= 2 * 120 - 3 * 75. \end{aligned}$$

(c) Menyelesaikan Persamaan Kongruen  $120x \equiv 75 \pmod{91}$

Periksa PBB(120, 91) = 1 (solusi ada).

Cari balikan 120 mod 91 menggunakan algoritma Euclidean, diperoleh balikan = 22.

Kalikan kedua sisi dengan 22:

$$x \equiv 22 * 75 \pmod{91} \rightarrow x \equiv 1650 \pmod{91}$$

$$1650 \pmod{91} = 12 \rightarrow x \equiv 12 \pmod{91}$$

$$\rightarrow \text{Solusi umum: } x = 12 + 91k, k \in \mathbb{Z}$$

2. Tentukan nilai  $p$  dan karakter uji dari kode ISBN berikut 9-2p301-777 jika diketahui bahwa  $5p \equiv 2 \pmod{3}$ , lalu tuliskan kembali kode ISBN nya.

(Nilai = 7,5)

**Jawaban:**

$$5p = 2 + 3k \rightarrow p = \frac{2+3k}{5}, \text{ untuk } k \text{ sembarang bilangan bulat}$$

Nili  $p$  yang memenuhi Untuk nilai  $k$ :

$$k = 1 \rightarrow p = 1$$

$$k = 2 \rightarrow p = 8/5 \text{ (} p \text{ tidak bulat)}$$

$$\dots$$

$$k = 6 \rightarrow p = 4$$

$$\dots$$

$$k = 11 \rightarrow p = 7$$

dst..

Nilai p yang memenuhi adalah 1,4, dan 7

Untuk mencari karakter uji, diketahui

$$\sum_{i=1}^9 ix_i \text{ mod } 11 = \text{karakter uji}$$

Untuk p = 1:

$$\begin{aligned} \sum_{i=1}^9 ix_i \text{ mod } 11 &= (1 * 9 + 2 * 2 + 3 * 1 + 4 * 3 + 5 * 0 + 6 * 1 + 7 * 7 + 8 * 7 + 9 * 7) \text{ mod } 11 \\ &= 202 \text{ mod } 11 = 4 \end{aligned}$$

Untuk p = 4:

$$\begin{aligned} \sum_{i=1}^9 ix_i \text{ mod } 11 &= (1 * 9 + 2 * 2 + 3 * 4 + 4 * 3 + 5 * 0 + 6 * 1 + 7 * 7 + 8 * 7 + 9 * 7) \text{ mod } 11 \\ &= 211 \text{ mod } 11 = 2 \end{aligned}$$

Untuk p = 7:

$$\begin{aligned} \sum_{i=1}^9 ix_i \text{ mod } 11 &= (1 * 9 + 2 * 2 + 3 * 7 + 4 * 3 + 5 * 0 + 6 * 1 + 7 * 7 + 8 * 7 + 9 * 7) \text{ mod } 11 \\ &= 220 \text{ mod } 11 = 0 \end{aligned}$$

Kode ISBN yang memenuhi

9-21301-777-4

9-24301-777-2

9-27301-777-0

3. Seorang peneliti menemukan tiga jenis perangkat yang menghasilkan sinyal dengan frekuensi yang berbeda-beda dan bersifat periodik. Periode masing-masing perangkat adalah 5 detik, 7 detik, dan 11 detik. Peneliti mencatat bahwa pada detik ke-3, detik ke-5, dan detik ke-7, masing-masing perangkat menghasilkan sinyal secara bersamaan. Peneliti ingin mengetahui detik ke berapakah ketiga perangkat akan menghasilkan sinyal secara bersamaan untuk pertama kalinya  
(Nilai = 7,5)

**Jawaban:**

Chinese Remainder Theorem. Ketiga perangkat akan menghasilkan sinyal secara bersamaan pada detik ke-x, di mana x memenuhi sistem kekongruenan berikut:

$$x \equiv 3 \pmod{5}$$

$$x \equiv 5 \pmod{7}$$

$$x \equiv 7 \pmod{11}$$

Langkah 1: Hitung modulus total

$$m = 5 \times 7 \times 11 = 385$$

Langkah 2: Hitung M1, M2, dan M3

$$M1 = 7 \times 11 = 77$$

$$M2 = 5 \times 11 = 55$$

$$M3 = 5 \times 7 = 35$$

Langkah 3: Temukan balikan (invers) dari M1, M2, dan M3 modulo masing-masing modulus

Balikan M1 (77) modulo 5:

$$77 \equiv 2 \pmod{5}. \text{ Balikan } 2 \text{ modulo } 5 \text{ adalah } 3 \text{ karena } 2 \times 3 \equiv 1 \pmod{5}.$$

Balikan M2 (55) modulo 7:

$$55 \equiv 6 \pmod{7}. \text{ Balikan } 6 \text{ modulo } 7 \text{ adalah } 6 \text{ karena } 6 \times 6 \equiv 1 \pmod{7}.$$

Balikan M3 (35) modulo 11:

$$35 \equiv 2 \pmod{11}. \text{ Balikan } 2 \text{ modulo } 11 \text{ adalah } 6 \text{ karena } 2 \times 6 \equiv 1 \pmod{11}.$$

Langkah 4: Hitung solusi x

$$x = (3 \times 77 \times 3) + (5 \times 55 \times 6) + (7 \times 35 \times 6)$$

$$x = (3 \times 77 \times 3) + (5 \times 55 \times 6) + (7 \times 35 \times 6)$$

$$x = 693 + 1650 + 1470$$

$$x = 3813$$

Langkah 5: Hitung x modulo m

$$x \equiv 3813 \pmod{385}$$

$$385 \times 9 = 3465$$

$$3813 - 3465 = 348$$

Jadi,  $x \equiv 348 \pmod{385}$ . Solusi terkecil adalah 348. Maka, ketiga perangkat akan menghasilkan sinyal secara bersamaan pertama kali pada detik ke-348

4. Sebuah acara makan malam di kampus akan diadakan dengan meja bundar yang dapat menampung 10 orang. Di antaranya terdapat 5 orang mahasiswa, 3 orang staf, dan 2 orang dosen (Rektor dan Kepala Jurusan). Berapa jumlah cara pengaturan duduk mereka jika: **(Nilai = 10)**
- Tidak ada batasan khusus
  - Semua mahasiswa harus duduk bersama, semua staf harus duduk bersama, dan semua dosen harus duduk bersama
  - Rektor dan Kepala Jurusan harus duduk di ujung meja yang berlawanan, dan mahasiswa serta staf harus duduk secara bergantian

**Jawaban:**

(a)  $(10 - 1)! = 9! = 362880$  cara

(b) Jika semua mahasiswa, staf, dan dosen harus duduk bersama, pertama-tama kelompok mereka dianggap sebagai satu kesatuan. Ada 3 kelompok (mahasiswa, staf, dosen), sehingga jumlah cara mengatur kelompok adalah  $(3 - 1)! = 2! = 2$ . Kemudian, pengaturan di dalam kelompok masing-masing:

- Mahasiswa:  $5! = 120$

- Staf:  $3! = 6$

- Dosen:  $2! = 2$

Jadi, total cara pengaturan adalah  $2 \times 120 \times 6 \times 2 = 2880$  cara

(c) Rektor dan Kepala Jurusan harus duduk di ujung meja yang berlawanan. Di meja bundar, ada 2 pilihan untuk posisi mereka (Rektor di posisi 1 dan Kepala Jurusan di posisi 10, atau sebaliknya). Sisa 8 orang (5 mahasiswa dan 3 staf) harus duduk secara bergantian.

Karena mereka harus bergantian, urutan duduk adalah Rektor, mahasiswa, staf, mahasiswa, staf, mahasiswa, staf, mahasiswa, Kepala Jurusan. Jumlah cara mengatur mahasiswa dan staf:

- Mahasiswa:  $5! = 120$
- Staf:  $3! = 6$

Jadi, total cara pengaturan adalah  $2 \times 120 \times 6 = 1440$  cara.

Catatan: Soal 4c tidak diujikan pada UAS ini

5. Hitunglah banyaknya kata yang berbeda yang bisa dibuat dengan hanya menggunakan huruf 'A' sebanyak  $N$  buah dan huruf 'B'  $M$  buah, dengan syarat tidak boleh ada 2 huruf 'B' yang bersebelahan dan semua huruf 'B' dan 'A' harus digunakan. (Nilai = 7,5)

(a) Jawablah dalam bentuk variabel  $N$  dan  $M$ , dengan asumsi  $N > M$ .

(b) Hitunglah jika  $N = 5$  dan  $M = 4$  (boleh dihitung manual, dengan ditunjukkan semua kombinasinya, jika jawaban soal a tidak ketemu)

Jawaban:

(a) Pertama-tama, kita buat susunan seperti ini



Awalnya diletakkan huruf 'B' sebanyak  $M$  buah, kemudian diantara huruf B ditaruh masing masing huruf A, sehingga total huruf A yang diletakkan sebanyak  $M-1$  buah. Hal ini dilakukan agar menjamin bahwa tidak ada 2 huruf 'B' yang bersebelahan.

Kemudian, sisanya, yaitu  $N - M + 1$  huruf 'A', akan diletakkan di sembarang tempat. Sehingga berdasarkan prinsip **kombinasi dengan pengulangan** dengan  $N - M + 1$  huruf 'A' dan  $M + 1$  kotak (diantara huruf B atau di sebelah paling kanan atau di sebelah paling kiri), didapatkan jumlah total caranya

$$C((N - M + 1) + (M + 1) - 1, (M + 1) - 1) = C(N + 1, M)$$

(b) Jika dihitung dengan rumus langsung didapatkan

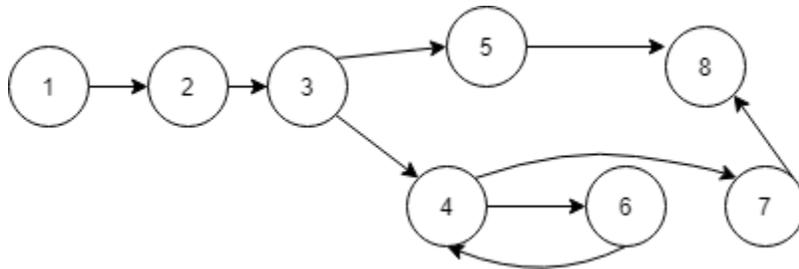
$$C(N+1, M) = C(5 + 1, 4) = C(6, 4) = 15$$

6. Terdapat kode program Python pada kotak sebelah kanan. Buatlah graf alir (*flow graph*) yang merepresentasikan alur program tersebut, dan sebutkan bagian kode yang direpresentasikan oleh tiap simpul (simpul diberi nomor dari 1, 2, dst. Beri keterangan simpul nomor sekian menyatakan kode yang mana).

(Nilai = 7,5)

```
x = input()
y = input()
if x > y:
    print(x+y)
else:
    while(x < y):
        x += 1
    print(x + y)
print('done')
```

**Jawaban:**

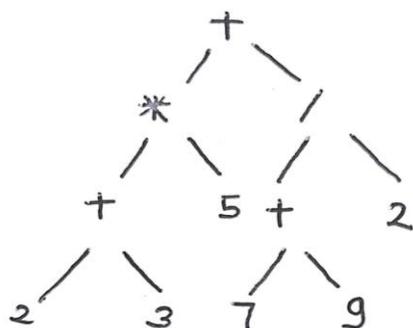


- 1 : x = input()
- 2 : y = input()
- 3 : if x>y
- 4 : while x < y:
- 5. print(x+y)
- 6 : x+=1
- 7 : print(x+y)
- 8 : print('done')

7. Gambar pohon biner dari ekspresi dalam bentuk *prefix* berikut: + \* + 2 3 5 / + 7 9 2. Lalu, tentukan hasilnya dan susun kembali menjadi ekspresi dalam bentuk *infix*!

(Nilai = 7,5)

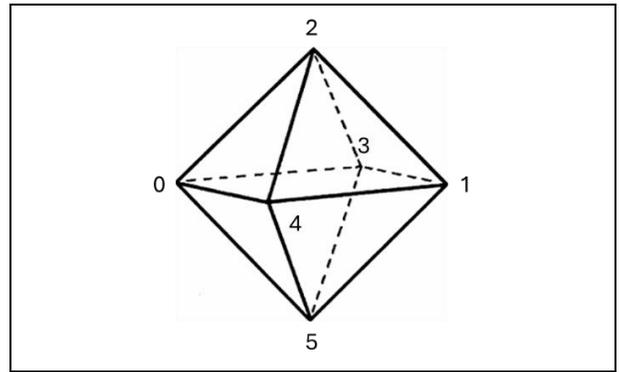
**Jawaban:**



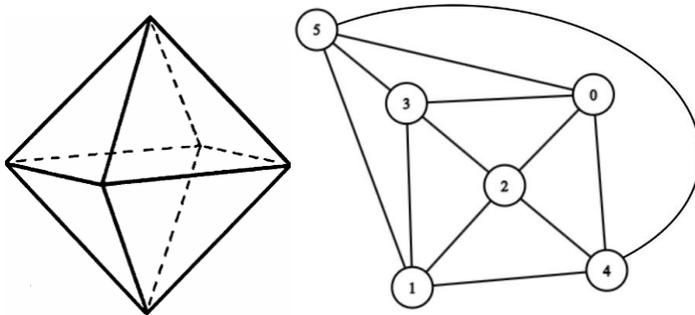
Hasil:  $(2 + 3) * 5 + (7 + 9) / 2 = 25 + 8 = 33$

Ekspresi infix:  $2 + 3 * 5 + 7 + 9 / 2$

8. Graf di kanan ini dinamakan graf oktahedron. Titik sudut oktahedron menyatakan sebuah bandara. Setiap bandara memiliki jalur penerbangan ke empat bandara lainnya. Tentukan apakah graf dari bandara-bandara tersebut planar! Selain itu, apabila seorang pilot terbang dari bandara X, apakah ia dapat melalui setiap jalur penerbangan tepat sekali dan berakhir kembali di bandara X?



**Jawaban:**



Berdasarkan deskripsi dari bandara-bandara tersebut, graf tersebut dapat disederhanakan menjadi sebuah oktahedron. Oktahedron tersebut bila digambarkan dalam sebuah bidang dua dimensi akan menjadi seperti gambar di sampingnya. Perhatikan bahwa tidak ada satupun upagraf yang isomorfik dengan graf Kuratowski sehingga **planar**.

Karena derajat setiap simpul adalah 4, maka sirkuit euler dapat dibentuk dari graf tersebut. Berdasarkan definisi dari sirkuit euler, maka seorang pilot yang terbang dari bandara X **dapat kembali** ke bandara X dengan melalui setiap jalur penerbangan tepat sekali.

9. Diberikan teks dengan distribusi karakter sebagai berikut:

**(Nilai = 10)**

Karakter	A	B	C	D	E	F	G
Frekuensi	25	15	10	8	7	3	2

- (a) Bangun pohon Huffman dan tentukan kode Huffman untuk setiap karakter.  
 (b) Hitung rasio kompresi menggunakan kode Huffman dibandingkan dengan kode biner dengan panjang tetap. Bandingkan hasilnya dengan kode ASCII yang menggunakan 7 bit per karakter

**Jawaban:**

(a) Pohon Huffman dan Kode Huffman:

Proses pembentukan pohon Huffman:

1. Gabungkan G dan F → GF (5)
2. Gabungkan E dan GF → EGF (12)
3. Gabungkan D dan C → DC (18)
4. Gabungkan B dan EGF → B\_EGF (27)
5. Gabungkan DC dan A → DC\_A (43)
6. Gabungkan B\_EGF dan DC\_A → root (70)

Kode Huffman:

- A: 0
- B: 10
- C: 110
- D: 1110
- E: 11110
- F: 111110
- G: 111111

(b) Rasio Kompresi:

- Kode Huffman: Rata-rata panjang bit =  $(25 \times 1 + 15 \times 2 + 10 \times 3 + 8 \times 4 + 7 \times 5 + 3 \times 6 + 2 \times 6) / 70 \approx 2.6$  bit/karakter
- Kode Biner Tetap: 3 bit/karakter (karena 7 karakter memerlukan 3 bit)
- Rasio Kompresi:  $(2.6/3) \times 100\% \approx 86.67\%$
- Bandingkan dengan ASCII: Jika menggunakan ASCII (7 bit/karakter), rasio kompresi =  $(2.6/7) \times 100\% \approx 37.14\%$

(c) Kompresi Pesan "ABECADFA":

Sebelum Kompresi (ASCII): 8 karakter  $\times$  7 bit = 56 bit

Setelah Kompresi (Huffman): A(1) + B(2) + E(5) + C(3) + A(1) + D(4) + F(6) + A(1) = 1+2+5+3+1+4+6+1 = 23 bit

Rasio Kompresi:  $(23/56) \times 100\% \approx 41.07\%$

10. Sebuah pesan terdiri dari karakter-karakter di dalam himpunan {D, I, K, R, S, T, U, \_}. Pesan tersebut sudah dikompresi dengan Metode Huffman. Kode Huffman untuk setiap karakter adalah sebagai berikut: (Nilai = 7,5)

D = 1110, I = 101, K = 100, R = 110, S = 010, T = 00, U = 011, \_ = 1111

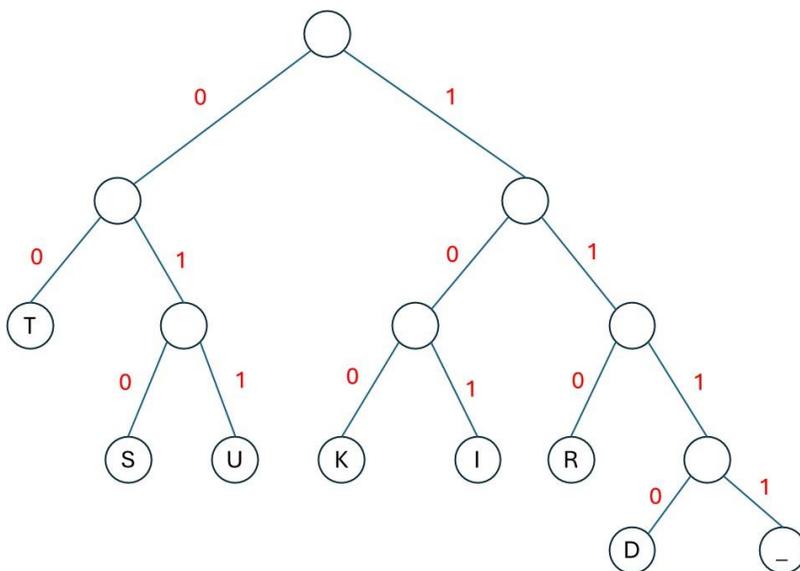
(a) Bentuklah pohon Huffman dari kode Huffman di atas. Pohon Huffman adalah sebuah *prefix code*.

(b) Lakukan *decoding* untuk string biner berikut:

0100011001110000011110111111010101010011010100

Jawaban:

(a)



(b) STRUKTUR\_DISKRIT

<p>11. Diberikan matriks persegi berukuran <math>n \times n</math>. Algoritma di sebelah kanan berikut memeriksa apakah matriks tersebut simetris dengan membandingkan setiap elemen <math>matrix[i][j]</math> dengan <math>matrix[j][i]</math>. Tentukan kompleksitas waktu algoritma dan kompleksitas waktu asimptotik algoritma ini dalam notasi Big-O.</p> <p style="text-align: right;">(Nilai = 7,5)</p>	<pre><b>function</b> isSymmetric(matrix: <math>n \times n</math>) <b>→ boolean</b> <b>Deklarasi</b>   simetris: <b>boolean</b> <b>Algoritma</b>   simetris <math>\leftarrow</math> <b>true</b>   <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>n</math> <b>do</b>     <b>for</b> <math>j \leftarrow i + 1</math> <b>to</b> <math>n</math> <b>do</b>       <b>if</b> <math>matrix[i][j] \neq matrix[j][i]</math> <b>then</b>         simetris <math>\leftarrow</math> <b>false</b>       <b>endif</b>     <b>endfor</b>   <b>endfor</b></pre>
--	---

**Jawaban:**

Kompleksitas waktu algoritma ini adalah  $O(n^2)$

Algoritma menggunakan dua loop bersarang:

1. Outer loop berjalan  $n$  kali ( $i$  dari 1 hingga  $n$ )
2. Inner loop untuk setiap  $i$ , berjalan  $n-i$  kali ( $j$  dari  $i+1$  hingga  $n$ )

Total iterasi adalah jumlah deret aritmetika:

$$\sum_{i=1}^{n-1} (n - i) = \frac{n(n-1)}{2} = O(n^2)$$

12. Tentukan apakah pernyataan berikut ini benar atau salah!

(Nilai = 8)

- a.  $n \log n + n/2$  adalah  $O(n)$
- b.  $\sqrt{n} + \log n$  adalah  $\Omega(1)$
- c.  $10\sqrt{n} + \log n$  adalah  $O(n)$
- d.  $\sqrt{n} + \log n$  adalah  $\Omega(\log n)$
- e.  $\sqrt{n} + \log n$  adalah  $\Theta(n)$
- f.  $2 + n + 4n^2$  adalah  $O(n^3)$
- g.  $n \log n + n$  adalah  $\Omega(n)$
- h. Diberikan  $T_1(n) = n$  dan  $T_2(n) = 5n^2$ , maka  $T_1(n)T_2(n) = O(n^3)$

Jawaban:

- a. Salah  
Karena  $n \log n$  lebih dominan dari pada  $n$  sehingga seharusnya  $O(n \log n)$
- b. Benar  
Karena Jika  $T(n) = \Omega(g(n))$  dengan  $g(n)$  adalah 1, maka akan terdapat tetapan  $C$  dan  $n_0$  sedemikian sehingga  $T(n) \geq C g(n)$ , untuk  $n \geq n_0$ . Hal ini karena waktu yang dominan ( $\sqrt{n}$ ), lebih cepat tumbuh dari 1
- c. Benar  
Karena  $0 \leq \sqrt{n} \leq n$  dan  $10\sqrt{n} \geq \log(n)$ , maka  $10\sqrt{n} + \log n \leq 10n + n = 11n$ , maka  $10\sqrt{n} + \log n = O(n)$
- d. Benar  
Karena Jika  $T(n) = \Omega(g(n))$  dengan  $g(n)$  adalah  $\log n$ , maka akan terdapat tetapan  $C$  dan  $n_0$  sedemikian sehingga  $T(n) \geq C g(n)$ , untuk  $n \geq n_0$ . Hal ini karena waktu yang dominan ( $\sqrt{n}$ ),

lebih cepat tumbuh dari  $\log n$ .

e. Salah

Karena waktu yang dominan yaitu  $\sqrt{n}$  tumbuh lebih pelan daripada  $n$ , sehingga jika  $\Omega$  dari persamaan adalah  $\Omega(n)$ , maka tidak terpenuhi

f. Benar, karena  $2 + n + 4n^2 \leq 7n^3$  untuk  $n \geq 1$ .

g. Benar, karena  $n \log n + n \geq 2n$  untuk  $n \geq 1$

h. Benar, karena  $T_1(n)T_2(n) = O(n \cdot 5n^2) = O(5n^3) = O(n^3)$

13. Apa prediksi (atau harapan) nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

(Nilai = 2)

**TOTAL NILAI = 102**