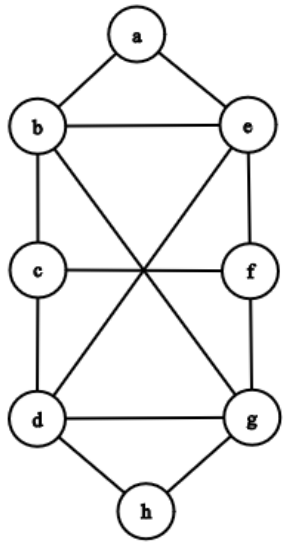
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Nama :…….....
NIM/Kelas :…….....
T.tangan :…….....

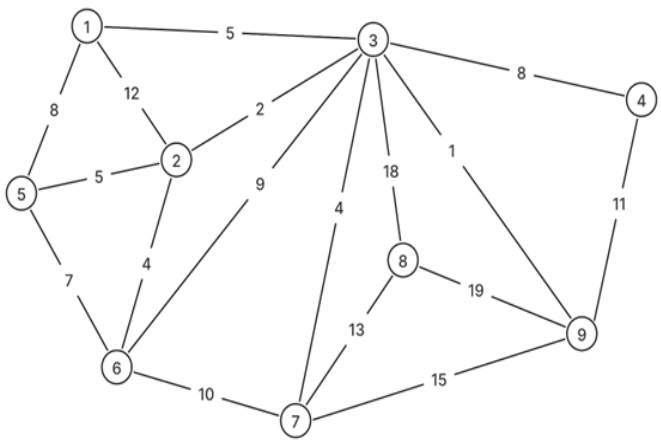
Solusi Kuis ke-4 IF1220 Matematika Diskrit (3 SKS) – Graf, Pohon, dan Kompleksitas Algoritma
Dosen: Rinaldi, Arrival Dwi Sentosa
Kamis, 5 Juni 2025
Waktu: 90 menit

1. Diberikan gambar sebuah graf G seperti pada **Gambar 1** di samping kanan ini.
   a. Tunjukkan dengan ketidaksamaan Euler bahwa graf G tidak planar.
   b. Tunjukkan dengan Teorema Kuratowski bahwa graf G tidak planar.
   c. Tentukan bilangan kromatik dari graf G.

   (Nilai: 15)



Gambar 1



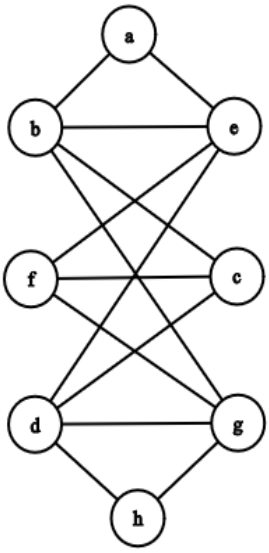
Gambar 2

Jawaban:

a. $n = 8, e = 13$

   Dengan menggunakan ketidaksamaan euler diperoleh
   $13 \geq 2(8) - 4 = 12$
   Dapat dilihat jika graf G tidak planar

b. Perhatikan bahwa graf tersebut isomorfik dengan graf G' ini.



   Perhatikan bahwa graf G' memiliki upagraf yang sama dengan K3,3 sehingga graf G tidak planar.

c. K3,3 merupakan graf bipartit (upagraf selain simpul a dan h) sehingga bilangan kromatik untuk upagraf ini adalah 2. Karena simpul a dan h tidak berhubungan maka tambahkan satu jenis warna tambahan agar tidak melanggar aturan. Sehingga bilangan kromatik graf G adalah 3.

2. Diberikan graf tak berarah dengan simpul $\{A, B, C, D, E, F, G, H\}$ dan sisi **(Nilai: 15)**
$(A,B), (A,C), (A,D), (B,C), (B,E), (B,F), (C,D), (C,G), (D,E), (D,H), (E,F), (E,G), (F,H), (G,H)\}$.
   a. Apakah graf ini Eulerian (mengandung sirkuit Euler)? Jika ya, berikan sirkuit Euler. Jika tidak, berikan alasan mengapa.
   b. Apakah graf ini Hamiltonian? Jika ya, berikan sirkuit Hamilton. Jika tidak, berikan alasan mengapa.
   c. Jika graf ini tidak Eulerian, tidak Hamiltonian, atau tidak keduanya, modifikasi graf dengan menambahkan sisi minimum agar menjadi Eulerian dan Hamiltonian.

Jawaban:

a. Cek derajat simpul
   - A terhubung ke B, C, D (derajat 3)
   - B terhubung ke A, C, E, F (derajat 4)
   - C terhubung ke A, B, D, G (derajat 4)
   - D terhubung ke A, C, E, H (derajat 4)
   - E terhubung ke B, D, F, G (derajat 4)
   - F terhubung ke B, E, H (derajat 3)
   - G terhubung ke C, E, H (derajat 3)
   - H terhubung ke D, F, G (derajat 3)

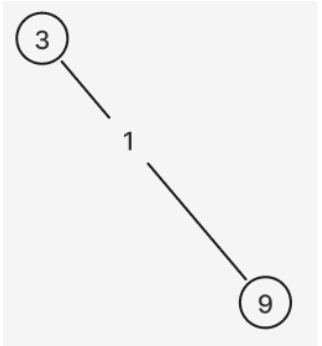
   Karena terdapat simpul yang berderajat ganjil (A, F, G, H) maka graf ini tidak memiliki sirkuit Euler dan bukan graf Eulerian.

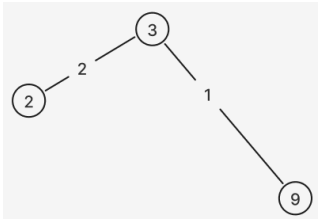
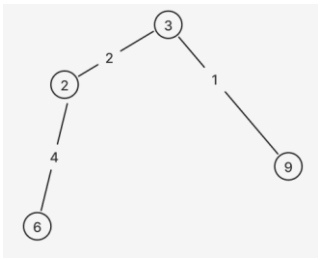
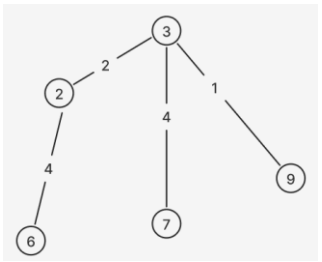
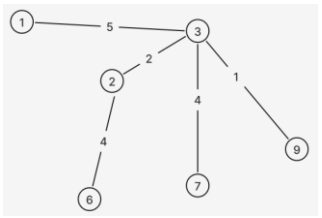
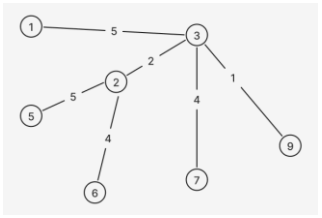
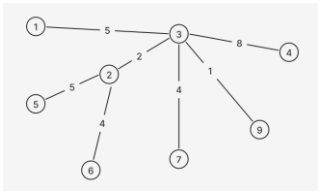
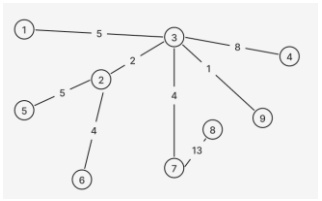
b. Ya, sirkuitnya adalah $\{(A,B), (B,E), (E,F), (F,H), (H,G), (G,C), (C,D), (D,A)\}$.

c. Hubungkan A dengan H dan F dengan G sehingga sisi baru yang muncul adalah (A,H) dan (F,G). Sirkuit Eulernya adalah (tidak perlu ada di jawaban, ini hanya untuk membuktikan bahwa graf adalah Eulerian) $\{(A,B), (B,E), (E,G), (G,F), (F,H), (H,D), (D,C), (C,A), (A,H),$
$(H,G), (G,C), (C,B), (B,F), (F,E), (E,D), (D,A)\}$

3. Tentukanlah pohon merentang minimum serta bobot total akhir dari graf pada **Gambar 2** di atas dengan menggunakan Algoritma Kruskal! Buatlah tabel yang memperlihatkan proses pembentukan pohon merentang minimum step by step (Jika terdapat sisi berbobot sama, pilihlah sisi yang memiliki simpul terkecil) **(Nilai: 15)**

Jawaban:

Iterasi	Sisi	Bobot	Total bobot Sementara	Pohon merentang
1	3-9	1	1	

2	2-3	2	3	
3	2-6	4	7	
4	3-7	4	11	
5	1-3	5	16	
6	2-5	5	21	
7	3-4	8	29	
8	7-8	13	42	
Bobot Total		42		

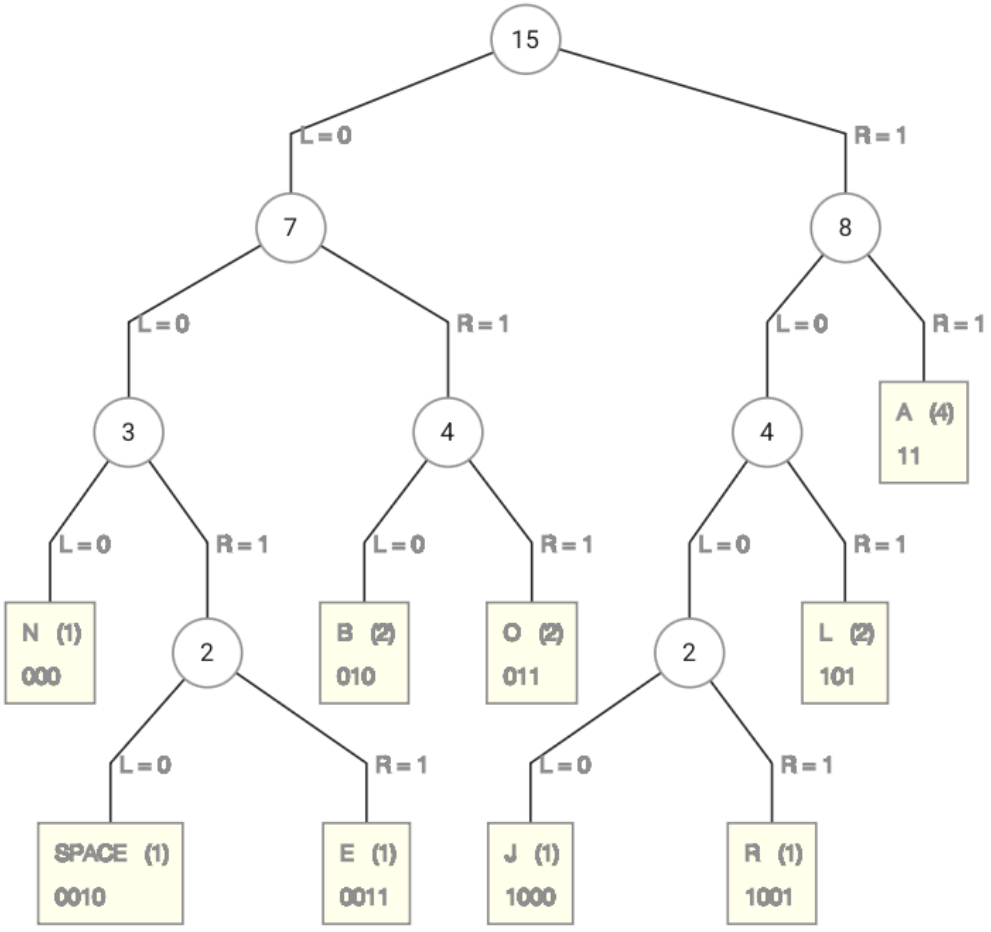
4. Terdapat sebuah pesan “ALJABAR BOOLEAN” dalam sebuah *script*. Berapakah panjang kode pesan tersebut jika dikodekan dengan kode Huffman (termasuk spasi)? Tanpa petik dua “ tidak termasuk pesan. **(Nilai: 15)**

Jawaban:

Frekuensi :

A	4	O	2
L	2	E	1
J	1	N	1
B	2	SPACE	1
R	1		

Huffman Tree:



Simbol	Frekuensi	Kode
A	4	11
L	2	101

J	1	1000
B	2	010
R	1	1001
O	2	011
E	1	0011
N	1	000
SPACE	1	0010

Maka panjang kode,

- A: frekuensi 4 × panjang kode 2 = 8 bit
- L: frekuensi 2 × panjang kode 3 = 6 bit
- J: frekuensi 1 × panjang kode 4 = 4 bit
- B: frekuensi 2 × panjang kode 3 = 6 bit
- R: frekuensi 1 × panjang kode 4 = 4 bit
- O: frekuensi 2 × panjang kode 3 = 6 bit
- E: frekuensi 1 × panjang kode 4 = 4 bit
- N: frekuensi 1 × panjang kode 3 = 3 bit
- SPACE: frekuensi 1 × panjang kode 4 = 4 bit

Total panjang kode = 8 + 6 + 4 + 6 + 4 + 6 + 4 + 3 + 4 = **45 bit**

5. Diberikan masukan berupa rangkaian karakter dengan urutan sebagai berikut: T, E, K, N, I, F, O, R, M, A, B, J, G
   a. Gambarkan pohon pencarian (*binary search tree*) yang terbentuk.
   b. Tentukan hasil penelusuran *preorder*, *inorder*, dan *postorder*, dari pohon jawaban (a) di atas. **(Nilai: 15)**

Jawaban:

Alur pembuatan pohon:

1. T → root
2. E < T → kiri T
3. K > E → kanan E
4. N > K → kanan K
5. I < N → kiri N
6. F < I → kiri I
7. O > N → kanan N
8. R > O → kanan O

9. M < N → kiri N → kiri I → kanan I → kanan I
10. A < E → kiri E → kiri A
11. B > A → kanan A
12. J > I → kanan I
13. G > F → kanan F

Hasil pohon

6. Diberikan dua algoritma berikut:

```
def p1(n):
    total = 1
    for i in range(1, n + 1):
        term = 1
        for _ in range(1, i + 1):
            term *= n
        total += term
    return total
```

```
def p2(n):
    result = 1
    for _ in range(n, 0, -1):
        result = 1 + result * n
    return result
```

Untuk sebarang nilai $n \geq 0$, kedua fungsi tersebut akan menghasilkan nilai yang sama. Dari kedua algoritma tersebut, tentukan: **(Nilai: 10)**

a. Jumlah operasi perkalian dan penjumlahan yang dilakukan kedua algoritma tersebut sebagai fungsi dari n!
b. Nyatakan kompleksitas waktu algoritma tersebut dalam notasi Big-O, Big-Ω, dan Big-Θ!

Jawaban:

a. p1(0) = 0 penjumlahan + 0 perkalian = 0

p1(1) = 1 penjumlahan + 1 perkalian = 2
p1(2) = 2 penjumlahan + 3 perkalian = 5
p1(3) = 3 penjumlahan + 6 perkalian = 9
…
p1(n) = $n + \frac{n}{2}(1+n) = \frac{3n}{2} + \frac{n^2}{2}$

p2(0) = 0 penjumlahan + 0 perkalian = 0

p2(1) = 1 penjumlahan + 1 perkalian = 2
p2(2) = 2 penjumlahan + 2 perkalian = 4
p2(3) = 3 penjumlahan + 3 perkalian = 6
…
p2(n) = 2n

b. p1:

$O(n^2)$ karena $\frac{3n}{2} + \frac{n^2}{2} \leq n^2$
$\Omega(n^2)$ karena $\frac{3n}{2} + \frac{n^2}{2} \geq \frac{n^2}{2}$
$\Theta(n^2)$ karena $O(n^2) = \Omega(n^2)$
p2:
$O(n)$ karena $2n \leq 4n$
$\Omega(n)$ karena $2n \geq n$
$\Theta(n)$ karena $O(n) = \Omega(n)$

7. Diketahui sebuah program Python dengan 4 buah fungsi sebagai berikut:

```
from typing import List

def get_factor_sums(num: int) -> int:
    factor_num = 0
    for i in range(1, math.sqrt(num)):
        if num % i == 0:
           factor_num += i
           if num // i != i:
              factor_num += (num // i)
    return factor_num
```

```
def triangular_sum(m: int, n: int) -> int:
    sume = 0
    for i in range(m):
        for j in range(n):
            sume += (i * m + j) + 1
    return sume
```

```
def right_shift_amount(num: int) -> int:
    amount = 0
    while (num > 0):
        num = num // 2
        amount += 1
    return amount
```

```
def sum_sum_tulang(num: int) -> int:
    sume = 0
    for i in range(num):
        block = int(i ** (1/3))
        for j in range(block):
            sume += j
    return sume
```

a. Tentukan notasi Big-O dan urutkan fungsi diatas dari yang paling mahal (paling lambat) sampai yang paling murah!
b. Berikan penjelasan mengapa `get_factor_sum` memiliki `for` dari 1 sampai `sqrt(num)` dan bukan `num` saja

(Nilai: 15)

Jawaban:

a. Kompleksitas:
   i. get_factor_sum = O(sqrt(n))
   ii. triangular_sum = O(n^2)
   iii. right_shift_amount O(log(n))
   iv. sum_sum_tulang = O(n * cbrt(n))

   Urutan = triangular_sum, sum_sum_tulang, get_factor_sum, right_shift_amount

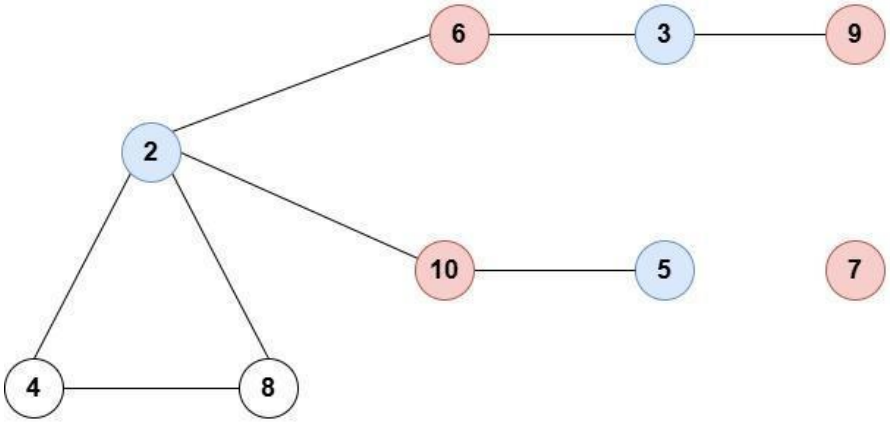
b. Karena ketika menentukan sebuah angka faktor dari angka input, jika angkanya benar, maka dia juga menjumlahkan angka lain yaitu angka input/angka yang didapat karena dipastikan itu adalah faktor dari

angka input itu juga. Untuk angka input/angka yang didapat yang hasilnya angka yang didapat itu sendiri tidak diikuti karena akan memasukkan angka yang sama

8. **(Bonus 5)** Diberikan bilangan asli 2 sampai 10. Buktikan bahwa tidak mungkin membagi himpunan bilangan ini menjadi 2 himpunan sehingga tidak ada bilangan yang dapat membagi bilangan lain di dalam satu himpunan yang sama. Gunakan teori graf untuk membuktikan pernyataan tersebut

Jawaban:

Buat graf dengan simpul yang melambangkan setiap bilangan dari 2 hingga 10. Sisi akan melambangkan pasangan bilangan di mana salah satu bilangan dapat membagi bilangan yang lain. Graf yang akan terbentuk adalah sebagai berikut.



Amati bahwa himpunan bilangan tersebut dapat dibagi ke dalam dua himpunan jika kita dapat mewarnai graf tepat dengan warna sehingga tidak ada simpul berhubungan dengan warna yang sama. Oleh karena itu, graf ini harus bipartit. Namun, perhatikan bahwa terdapat siklus 2-4-8. Siklus ini terdiri atas 3 simpul, di mana pewarnaan graf hanya dengan 2 warna tidak dapat dilakukan. Jika kita mewarnai simpul 4 dengan warna merah, maka 8 akan diwarnai dengan warna biru. Tetapi, hal ini tidak dapat dilakukan karena simpul 2 dan 8 berhubungan dan simpul 2 sudah berwarna biru. Dengan kata lain, graf ini bukan graf bipartit.
Jadi, terbukti bahwa kita tidak bisa membagi bilangan 2-10 menjadi 2 himpunan yang diminta soal.

9. **(Bonus 5)** Diberikan beberapa waktu proses T(n) dari 5 algoritma. Nyatakan ekspresi tersebut dalam notasi Big-O dan urutkan notasi Big-O tersebut dari yang tercepat!
   1) $T(n) = 10n\ logn\ +\ n^2$
   2) $T(n) = 2^n + n^5$
   3) $T(n) = log^2 n + n$
   4) $T(n) = n!$
   5) $T(n) = nlogn + n^3$

Jawaban:

a. $O(n^2)$

b. $O(2^n)$

c. $O(n)$

d. $O(n!)$

e. $O(n^3)$

Urutan: c - a - e - b - d

Total nilai: 100 + bonus 10 = 110

==========

Kerjakan mulai dari halaman dibalik ini, lalu pada kertas tambahan. Jika masih kurang, silakan pakai kertas sendiri