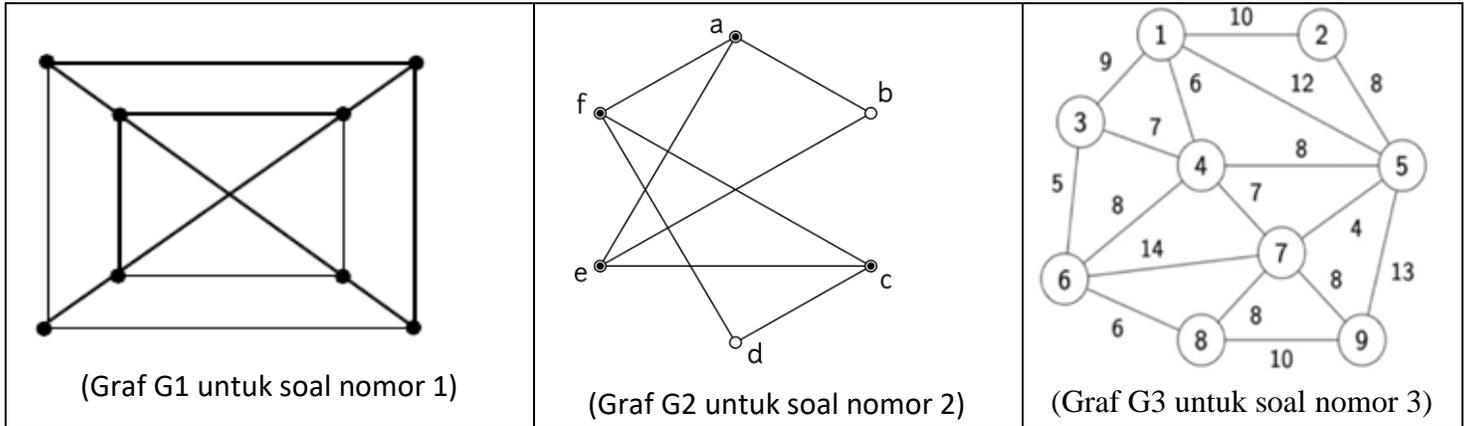


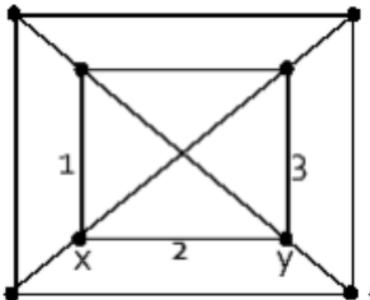
Solusi-Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Graf, Pohon, Kompleksitas Algoritma
 Dosen: Rinaldi Munir, Fariska Zakhralatifa, Nur Ulfa Maulidevi, Monterico Adrian
 Rabu, 29 November 2023
 Waktu: 55 menit



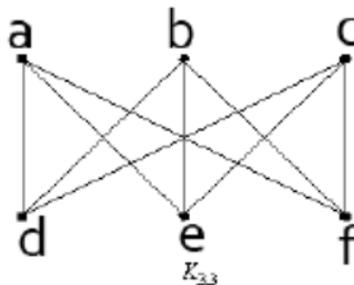
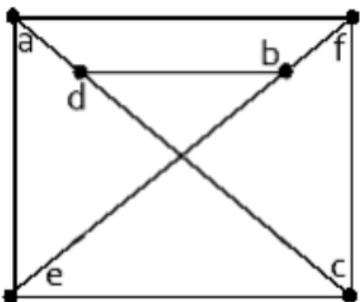
1. Apakah graf G1 pada gambar di atas merupakan graf planar? Jika ya, gambarkanlah dalam bentuk planar. Jika tidak planar, buktikan dengan teorema Kuratowski.

Jawaban:

Graf di atas bukanlah merupakan graf yang planar. Dengan Teorema Kuratowski, dapat ditunjukkan bahwa terdapat graf G' yang homeomorfik dengan graf G tersebut, dimana graf G' isomorfik dengan $K_{3,3}$.



Sebuah graf baru G' yang homeomorfik dengan graf awal dapat diperoleh dengan menghapus simpul X dan Y .



Graf baru ini isomorfik dengan graf $K_{3,3}$.

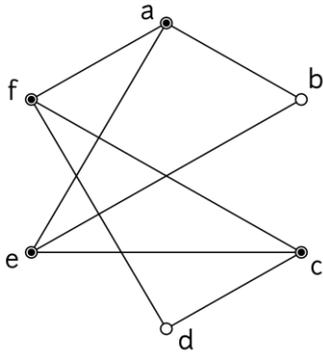
Berdasarkan teorema Kuratowski, karena graf memiliki upagraf homeomorfik yang isomorfik dengan $K_{3,3}$, maka graf tersebut tidaklah planar.

2. Diberikan graf G2 pada gambar di atas.
 a. Tentukan apakah graf tersebut memiliki lintasan hamilton? Jika ya, gambarkan lintasan hamilton pada

graf tersebut

- b. Tentukanlah sisi mana yang dapat diputus agar graf tersebut memiliki lintasan euler. Gambarkan lintasan euler dari graf yang telah diputus sisinya tersebut.

Jawaban:



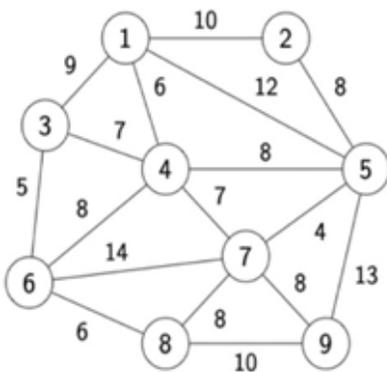
- a. Ya, salah satu lintasan hamilton pada graf tersebut adalah $b - e - a - f - d - c$
(jawaban-jawaban lain yang benar juga bisa diterima)
- b. Kita dapat memotong sisi $c-f$, sehingga dapat dibentuk lintasan euler sebagai berikut.
 $a - e - b - a - f - d - c - e$

Atau kita dapat memotong sisi $c-e$, sehingga dapat dibentuk lintasan euler sebagai berikut.

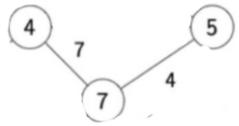
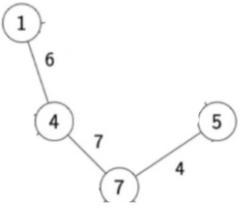
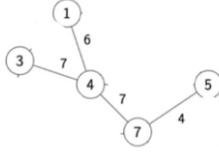
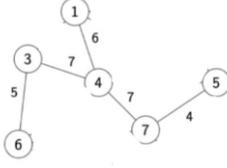
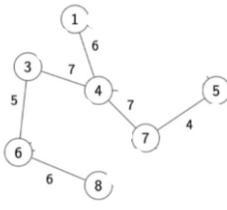
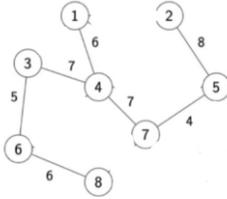
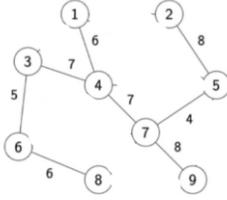
$a - e - b - a - f - d - c - f$
(jawaban-jawaban lain yang benar juga bisa diterima)

3. Tentukanlah **pohon merentang minimum** serta **bobot total akhir** dari graf G_3 pada gambar di atas dengan menggunakan Algoritma Prim! Buatlah tabel yang memperlihatkan proses pembentukan pohon merentang minimum step by step (Jika terdapat sisi berbobot sama, pilihlah sisi yang memiliki simpul terkecil)

Jawaban:



Iterasi	Sisi	Bobot	Pohon merentang
1	(5,7)	4	

2	(4,7)	7	
3	(1,4)	6	
4	(3,4)	7	
5	(3,6)	5	
6	(6,8)	6	
7	(2,5)	8	
8	(7,9)	8	
Bobot Total		51	

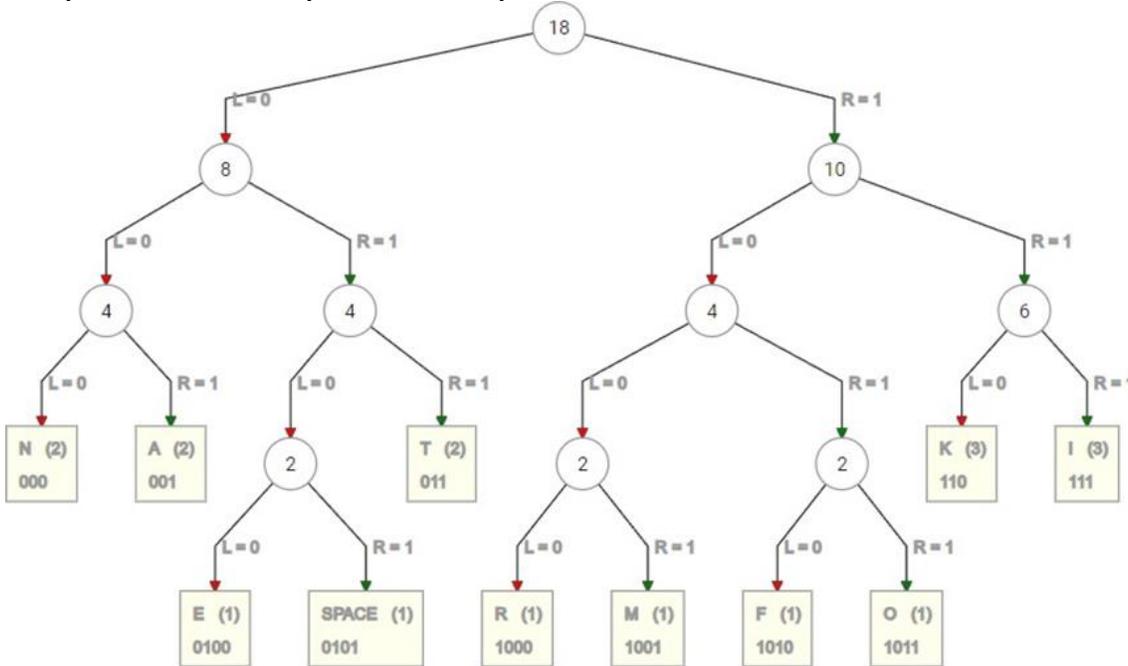
4. Terdapat sebuah pesan "TEKNIK INFORMATIKA" dalam sebuah script. Berapakah panjang kode pesan tersebut jika dikodekan dengan kode Huffman (termasuk spasi)!

Jawaban:

Hitung tiap frekuensi huruf unik dalam data tersebut, "TEKNIK INFORMATIKA"

T : 2 O : 1
 E : 1 R : 1
 K : 3 M : 1
 N : 2 A : 2
 I : 3 _ : 1
 F : 1

Jadi pohon Huffman dapat dibentuk seperti berikut,



Simbol	Frekuensi	Kode
T	2	011
E	1	0100
K	3	110
N	2	000
I	3	111
F	1	1010
O	1	1011
R	1	1000
M	1	1001
A	2	001
_	1	0101

Jadi panjang dari kode Huffman dari "TEKNIK INFORMATIKA" adalah,

$$2 * 3 + 1 * 4 + 3 * 3 + 2 * 3 + 3 * 3 + 1 * 4 + 1 * 4 + 1 * 4 + 1 * 4 + 2 * 3 + 1 * 4 = 60$$

5. Diberikan beberapa waktu proses $T(n)$ dari 5 buah algoritma (1, 2, 3, 4, 5). Nyatakan ekspresi tersebut dalam notasi Big-O dan urutkan notasi Big-O tersebut dari yang tercepat!
- $10\log(n) + (\log(n))^3$
 - $n + n^{1.23} + n^{1.24} + n^{1.25}$
 - $100000 + 10n^2 + n\sqrt{n}$
 - $999n(\log(n))^2 + 997n^2(\log(n))$
 - $25 + n(10 + \log(n))$

Jawaban:

Nomor	$T(n)$	$O(f(n))$
1	$10\log(n) + (\log(n))^3$	$O(\log(n))$
2	$n + n^{1.23} + n^{1.24} + n^{1.25}$	$O(n^{1.25})$
3	$100000 + 10n^2 + n\sqrt{n}$	$O(n^2)$
4	$999n(\log(n))^2 + 997n^2(\log(n))$	$O(n^2\log(n))$
5	$25 + n(10 + \log(n))$	$O(n\log(n))$

Urutan dari yang tercepat: 1 - 5 - 2 - 3 - 4

6. Diberikan cuplikan kode Python seperti berikut. Kompleksitas algoritma dihitung dari banyaknya operasi penjumlahan dan perkalian di dalam potongan program tersebut.

```
def unknown(n, a, b):
    result = 0

    for i in range(1, n):
        for j in range(1, i+1):
            result = a[i] * (b[j] + result)

    return result
```

- Tentukan $T(n)$
- Tentukan kompleksitas algoritma dalam notasi Big-O, Big-Omega, dan Big-Theta

Jawaban:

- Perhatikan bahwa tiap banyaknya iterasi yang dilakukan adalah $2 + 3 + 4 + \dots + (n + 1)$. Ini didapat dari loop for j yang melakukan iterasi sebanyak 2 kali untuk $i = 1$, 3 kali untuk $i = 2$, dan seterusnya. Jadi, dilakukan 1 buah penjumlahan dan perkalian sebanyak $2 + 3 + 4 + \dots + (n + 1)$. Banyaknya

penjumlahan dan perkalian yang dilakukan dapat dihitung menggunakan rumus deret aritmatika. Berikut banyaknya penjumlahan yang dilakukan.

$$\begin{aligned} a &= 2, b = 1 \\ S_n &= \frac{n}{2} (2a + (n-1)b) \\ S_n &= \frac{n}{2} (4 + n - 1) \\ S_n &= \frac{n}{2} (3 + n) \\ S_n &= \frac{n^2}{2} + \frac{3n}{2} \end{aligned}$$

Karena banyaknya penjumlahan dan banyaknya perkalian sama, maka didapat hasil akhir berupa
 $T(n) = n^2 + 3n$

b. Big-O:

$$T(n) = O(n^2) \text{ karena } n^2 + 3n \leq 4n^2$$

Big-Omega:

$$T(n) = \Omega(n^2) \text{ karena } n^2 + 3n \geq n^2$$

Big-Theta:

$$T(n) = \Theta(n^2) \text{ karena } O(n^2) \text{ dan } \Omega(n^2)$$