Program Studi Teknik Informatika Nama :…………………………

Sekolah Teknik Elektro dan Informatika NIM/Kelas :…………………………

Institut Teknologi Bandung T.tangan :…………………………

Solusi-Kuis ke-4 IF2120 Matematika Diskrit (3 SKS) – Graf, Pohon, Kompleksitas Algoritma

Dosen: Rinaldi Munir, Fariska Zakhralativa, Nur Ulfa Maulidevi, Monterico Adrian

Rabu, 29 November 2023

Waktu: 55 menit

|  |  |  |
| --- | --- | --- |
| (Graf G1 untuk soal nomor 1) | (Graf G2 untuk soal nomor 2) | (Graf G3 untuk soal nomor 3) |

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 ditunjukan bahwa terdapat graf G' yang homeomorfik dengan graf G tersebut, dimana graf G' isomorfik dengan K3,3.

A diagram of a square with a cross

Description automatically generated

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

A diagram of a square with a cross

Description automatically generated

Graf baru ini isomorfik dengan graf K3,3.

Berdasarkan teorema Kuratowski, karena graf memiliki upagraf homeomorfik yang isomorfik dengan K3,3, maka graf tersebut tidaklah planar.

1. Diberikan graf G2 pada gambar di atas.
2. Tentukan apakah graf tersebut memiliki lintasan hamilton? Jika ya, gambarkan lintasan hamilton pada graf tersebut
3. Tentukanlah sisi mana yang dapat diputus agar graf tersebut memiliki lintasan euler. Gambarkan lintasan euler dari graf yang telah diputus sisinya tersebut.

**Jawaban:**

A black background with white circles

Description automatically generated

1. Ya, salah satu lintasan hamilton pada graf tersebut adalah

b - e - a - f - d - c

(jawaban-jawaban lain yang benar juga bisa diterima)

1. Kita dapat memutus sisi c-f, sehingga dapat dibentuk lintasan euler sebagai berikut.

a - e - b - a - f - d - c - e

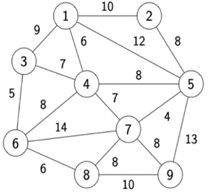
Atau kita dapat memutus sisi c-e, sehingga dapat dibentuk lintasan euler sebagai berikut.

a - e - b - a - f - d - c - f

(jawaban-jawaban lain yang benar juga bisa diterima)

1. Tentukanlah **pohon merentang minimum** serta **bobot total akhir** dari graf G3 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 | A line with numbers and circles  Description automatically generated |
| 2 | (4,7) | 7 | A diagram of a connection  Description automatically generated with medium confidence |
| 3 | (1,4) | 6 | A diagram of a network  Description automatically generated with medium confidence |
| 4 | (3,4) | 7 | A diagram of a network  Description automatically generated |
| 5 | (3,6) | 5 | A diagram of a network  Description automatically generated |
| 6 | (6,8) | 6 | A diagram of a network  Description automatically generated |
| 7 | (2,5) | 8 | A diagram of a network  Description automatically generated |
| 8 | (7,9) | 8 | A diagram of a network  Description automatically generated |
| Bobot Total | | 51 | |

1. 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,

A diagram of a tree

Description automatically generated

A black and white table with white text

Description automatically generated

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

1. 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!

**Jawaban:**

|  |  |  |
| --- | --- | --- |
| Nomor | T(n) | O(f(n) |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

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

1. 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 |

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

**Jawaban:**

1. Perhatikan bahwa tiap banyaknya iterasi yang dilakukan adalah 2 + 3 + 4 + ⋯ + (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 + ⋯ + (𝑛 + 1). Banyaknya penjumlahan dan perkalian yang dilakukan dapat dihitung menggunakan rumus deret aritmatika. Berikut banyaknya penjumlahan yang dilakukan.

A black background with white text

Description automatically generated

Karena banyaknya penjumlahan dan banyaknya perkalian sama, maka didapat hasil akhir berupa 𝑇(𝑛) =

𝑛^2 + 3𝑛

1. Big-O:

𝑇(𝑛) = 𝑂(𝑛^2) karena 𝑛^2 + 3𝑛 ≤ 4𝑛^2

Big-Omega:

𝑇(𝑛) = 𝑂𝑚𝑒𝑔𝑎(𝑛^2) karena 𝑛^2 + 3𝑛 ≥ 𝑛^2

Big-Theta:

𝑇(𝑛) = 𝑇ℎ𝑒𝑡𝑎(𝑛^2) karena 𝑂(𝑛^2) dan 𝑂𝑚𝑒𝑔𝑎(𝑛^2)