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

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

**(Shortlist)** Kuis ke-4 IF2153 Matematika Diskrit (3 SKS) – Pohon dan Kompleksitas Algoritma
Dosen: Bapak Rinaldi Munir, Ibu Harlili
Kamis, 14 Desember 2006
Waktu: 50 menit

1. **(Fajrin)** Misalkan *n* bilangan antara 100 dan 300 sedemikian hingga pohon *m-ary* dengan *n* buah simpul adalah pohon *m-ary* penuh dengan kedalaman 3 (asumsi bahwa pohon dengan hanya satu simpul memiliki kedalaman 0). Tentukan dua nilai yang mungkin untuk *n*. **(25)**

   **Jawab :**

   Jumlah simpul pada pohon *m-ary* penuh dengan kedalaman 3 adalah $\frac{m^4-1}{m-1}$. Untuk :

   $$m = 4 \rightarrow \frac{4^4-1}{4-1} = 85$$

   $$m = 5 \rightarrow \frac{5^4-1}{5-1} = 156$$

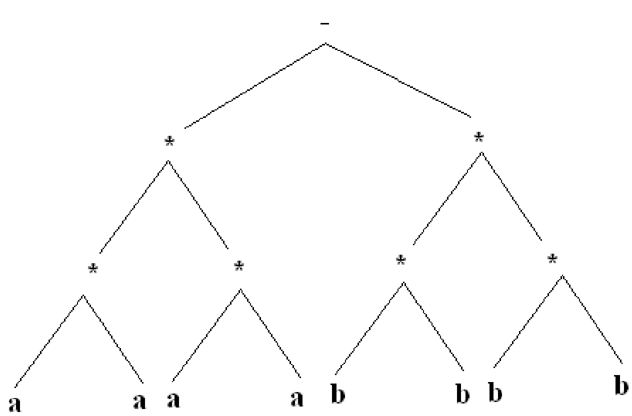
   $$m = 6 \rightarrow \frac{6^4-1}{6-1} = 259$$

   $$m = 7 \rightarrow \frac{7^4-1}{7-1} = 400$$

   Jadi, dua nilai yang mungkin untuk *n* adalah 156 dan 259.

2. **(Fajrin)** Gambarkan pohon ekspresi dengan kedalaman paling minimum untuk $a^4 - b^4$ dengan tidak menggunakan simpul ^ (pangkat), jadi hanya menggunakan simpul *a*, *b*, *, dan –. **(15)**

   **Jawab :**



3. **(Fajrin)** Berapakah nilai kompleksitas waktu asimptotik dalam notasi *O*-Besar untuk

$$T(n) = \begin{cases} k, & n = 1 \\ k + k^2 T(n-1), & n > 1 \end{cases}$$ **(20)**

**Jawab :**

$T(n) = k + k^2 T(n-1)$

$= k + k^2(k + k^2 T(n-2)) = k + k^3 + k^4 T(n-2)$

$= k + k^3 + k^4(k + k^2 T(n-3)) = k + k^3 + k^5 + k^6 T(n-3)$

$\vdots$

$= (k + k^3 + k^5 + \ldots + k^{2n-3}) + k^{2n-2} T(1)$

$= k + k^3 + k^5 + \ldots + k^{2n-3} + k^{2n=1}$

$= \dfrac{k(k^{2n} - 1)}{k^2 - 1} = O(k^{2n-1})$

4. **(Fajrin)** Buktikan atau sangkal kesamaan berikut :

a. $2n^2 - 100 = O(n)$ **(10)**

b. $n^3 2^n + n^2 3^n = O(n^3 2^n)$ **(10)**

**Jawab :**

a. $2n^2 - 100 = O(n)$ tidak benar, karena tidak mungkin terdapat *k* sehingga $2n^2 - 100 \le kn$ untuk setiap *n* yang lebih besar dari suatu $n_0$. Hal ini disebabkan $n(2n - k) \le 100$ tidak akan berlaku untuk nilai *n* yang jauh besar daripada *k*.

b. $n^3 2^n + n^2 3^n = O(n^3 2^n)$ tidak benar, karena tidak mungkin terdapat *k* sehingga $n^3 2^n + n^2 3^n \le kn^3 2^n$ untuk setiap *n* yang lebih besar dari suatu $n_0$. Hal ini disebabkan $3^n \le (k-1)n2^n$ tidak benar untuk nilai *n* yang besar.

5. **(Zakka)** a. Tentukan kompleksitas waktu dari algoritma dibawah ini jika melihat banyaknya jumlah proses `a ← a + 1` **(20)**

```
for i ← 1 to n do
   for j ← 1 to i do
      for k ← j to n do
         a ← a + 1
      endfor
   endfor
endfor
```

b. Tentukan pula nilai *O*-besar, *Ω*-besar, dan *Θ*-besar dari algoritma diatas (harus penjelasan) **(10)**

Solusi:
a.
Untuk $i = 1$,
Untuk $j = 1$, jumlah perhitungan = $n$ kali
Untuk $i = 2$,
Untuk $j = 1$, jumlah perhitungan = $n$ kali
Untuk $j = 2$, jumlah perhitungan = $n - 1$ kali
...
Untuk $i = n$,
Untuk $j = 1$, jumlah perhitungan = $n$ kali
Untuk $j = 2$, jumlah perhitungan = $n - 1$ kali
...
Untuk $j = n$, jumlah perhitungan = 1 kali.

Jadi jumlah perhitungan = $T(n) = n^2 + (n-1)^2 + (n-2)^2 + ... + 1$

b.
$O(n^3) = \Omega(n^3) = \Theta(n^3)$.

Salah satu cara penjelasan:
$T(n) = n^2 + (n-1)^2 + (n-2)^2 + ... + 1 = n(n+1)(2n+1)/6 = 2n^3 + 3n^2 + 1$.
Diperoleh $T(n) \leq 3n^3$ untuk $n \geq 4$ dan $T(n) \geq 2n^3$ untuk $n \geq 1$.

6. **(Zakka)** Adakah suatu algoritma yang menyebabkan nilai dari $\Theta$-besar tidak ada? Jika ada, berikan algoritmanya, jika tidak, buktikan. **(15)**

Solusi:
Tidak ada. Penjelasan, cukup mudah.

7. **(Zakka)** Perlihatkan bahwa $n^2 = O(2^n)$ tapi tidak berlaku sebaliknya. **(20)**

Solusi:
Bukti untuk $n^2 = O(2^n)$. Diperoleh $n^2 \leq 2^n$ untuk semua $n \geq 4$.
Bukti bahwa $2^n \neq O(n^2)$. Tidak akan ada konstanta $C$ sehingga $2^n \leq Cn^2$ untuk semua $n \geq n_0$.

8. **(Zakka)** Manakah yang lebih efektif, fungsi mencari nilai faktorial suatu bilangan dengan metode iteratif atau metode rekursif. Berikan kompleksitas waktu dari kedua algoritma dan penjelasan secukupnya. **(20)**

Solusi:
Tergantung penjelasan.

9. **(Ilham) Tunjukkan bahwa untuk semua formula dalam notasi prefiks yang terdiri dari himpunan simbol dan himpunan operator biner mengandung tepat**

**satu lebih banyak simbol daripada operator. (Kaitkan analisis anda dengan prinsip pohon). (25)**

**Jawab :**

Karena formula tersebut terdiri dari himpunan operator biner, maka dapat direpresentasikan menggunakan pohon biner dengan daunnya adalah simbol dan simpul yang bukan daun adalah operatornya.
*Untuk pohon biner penuh* :

Jumlah seluruh simpul sebuah pohon biner penuh dengan kedalaman $h$ adalah $2^{h+1} - 1$ , sedangkan jumlah daunnya adalah $2^h$. Oleh karena itu, jumlah simbol suatu formula jika direpresentasikan menggunakan pohon biner adalah $2^h$, sedangkan jumlah operatornya adalah $(2^{h+1} - 1) - 2^h = 2.2^h - 1 - 2^h = 2^h - 1$. Maka, terbukti bahwa jumlah simbol tepat satu lebih banyak dari jumlah operator.

*Untuk pohon biner tidak penuh* :

Pohon biner tidak penuh jika daun pada level-$h$ tidak sejumlah $2^h$. Dalam hal ini, level yang tidak lengkap tersebut, misalkan $h$, maka pada level sebelumnya yakni $h$-1 merupakan pohon biner penuh. Perubahan daun dari level sebelumnya menjadi akar mengakibatkan penambahan 1 buah daun dan 1 buah akar, sehingga perbedaan jumlah keduanya tidak akan berubah.

10. **(Ilham) Buktikan bahwa pohon penuh dengan jumlah cabang $m$ dengan $n$ simpul memiliki $i = (n - 1) / m$ simpul dalam dan $l = [(m - 1)n + 1] / m$ daun. (20)**

**Jawab :**

Anggap $n$ sebagai jumlah simpul, $i$ sebagai jumlah simpul dalam dan $l$ sebagai jumlah daun. Karena setiap $i$ simpul dalam memiliki $m$ buah anak, maka akan ada $mi$ buah simpul selain akar, oleh karena itu $n = mi + 1$, sehingga $\boldsymbol{i = (n - 1) / m}$ . $n$ juga bisa didapatkan dari jumlah daun $l$ ditambah dengan jumlah simpul dalam $i$, yakni $n = l + i$ . Dari formula ini didapatkan $\boldsymbol{l = n - i = n - (n - 1) / m = [(m - 1)n + 1] / m}$ (Terbukti).

11. **(Ilham) Berikan estimasi Big-$O$ untuk $f(n) = 3n \log(n!) + (n^2 + 3) \log n$, dimana $n$ adalah bilangan bulat positif. (25)**

**Jawab :**

Untuk C = 1 dan k = 1, $\log n! \leq \log n^n = n \log n$, maka $\log n!$ adalah $O(n \log n)$, sedangkan $3n$ adalah $O(n)$, sehingga $3n \log (n!)$ adalah $O(n^2 \log n)$.
Lalu, untuk $(n^2 + 3)\log n$, karena $n^2 + 3 < 2n^2$ ketika $n > 2$, maka $(n^2 + 3)$ adalah $O(n^2)$ dan menurut teorema maka $(n^2 + 3)\log n$ adalah $O(n^2 \log n)$.

Dengan menggunakan teorema, maka $3n \log(n!) + (n^2 + 3) \log n$ adalah $O(n^2 \log n)$.

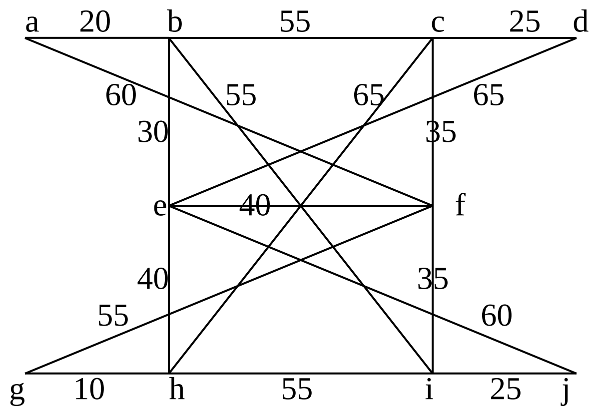
12. **(Ilham) Berikan estimasi Big-$O$ untuk $f(x) = (x + 1) \log (x^2 + 1) + 3x^2$ .** **(10)**

**Jawab :**

$(x + 1)$ adalah $O(x)$ dan $(x^2 + 1) \leq 2x^2$ ketika $x > 1$, sehingga
$\log (x^2 + 1) \leq \log (2x^2) = \log 2 + \log x^2 = \log 2 + 2 \log x \leq 3 \log x$, jika $x > 2$.
Hal ini menunjukkan $\log(x^2 + 1)$ adalah $O(\log x)$.

Dengan menggunakan teorema, maka $(x + 1) \log (x^2 + 1)$ adalah $O(x \log x)$. Karena $3x^2$ adalah $O(x^2)$, maka $(x + 1) \log (x^2 + 1) + 3x^2$ adalah $O(x^2)$.

13. **(Mira)**

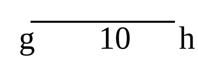
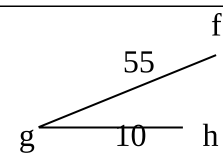



Carilah pohon merentang dari graf di atas dengan menggunakan (tunjukkan langkah-langkah pembuatannya) :

a. Algoritma Prim **(20)**
b. Algoritma Kruskal **(20)**

Jawab :

a.

| Langkah | Sisi | Bobot | Pohon Merentang |
|---|---|---|---|
| 1 | (g,h) | 10 |   |
| 2 | (f,g) | 55 |   |

Dst . .

b.

| sisi | (g,h) | (a,b) | (c,d) | (i,j) | (b,e) | (c,f) | (f,i) | (e,f) | (e,h) | (b,i) | (h,i) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| bobot | 10 | 20 | 25 | 25 | 30 | 35 | 35 | 40 | 40 | 55 | 55 |

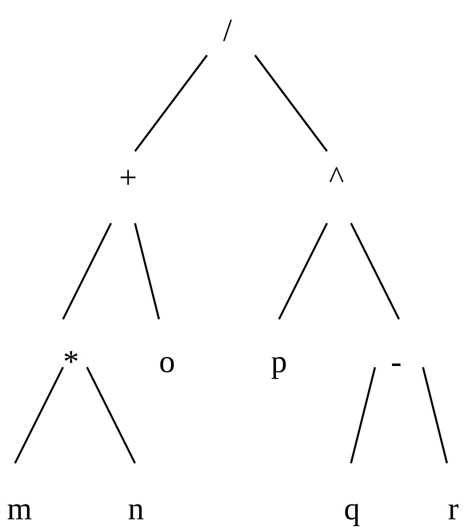
| (f,g) | (a,f) | (e,j) | (c,h) | (d,e) |
|---|---|---|---|---|
| 55 | 60 | 60 | 65 | 65 |

| Langkah | Sisi | Bobot | Pohon Merentang |
|---|---|---|---|
| 0 | | | a b c d e f g h i j |
| 1 | (g,h) | 10 | g 10 h |
| 2 | (a,b) | 20 | g 10 h<br>a 20 b |

Dst . .

14. **(Mira)** Buatlah pohon biner dari ekspresi aritmatik (notasi postfix) m n * o + p q r - ^ / , lalu nyatakan ekspresi tersebut dalam bentuk notasi lainnya (dengan 2 notasi) **(20)**

Jawab :




Infix : (m*n + o) / (p^ (q-r))

Prefix : / + * m n o ^ p - q r

15. **(Mira)** Apa definisi dari pohon ? **(10)**

Jawab :

Graf tak berarah terhubung yang tidak mengandung sirkuit.
Ada 2 keyword : terhubung dan tidak mengandung sirkuit

16. **(Mira)** Tentukan kode Huffman untuk string " hypochondriac". Syarat pada pohon Huffman yang dibangun ialah simbol dengan peluang lebih kecil sebagai anak kiri dan simbol dengan peluang lebih besar sebagai anak kanan, sisi kiri dilabeli dengan 0 dan sisi kanan dilabeli dengan 1. **(15)**

Jawab :

| Simbol | Kekerapan | Peluang | Kode Huffman |
|---|---|---|---|
| h | 2 | 2 / 13 | |
| y | 1 | 1 / 13 | |
| p | 1 | 1 / 13 | |
| o | 2 | 2 / 13 | |
| c | 2 | 2 / 13 | |
| n | 1 | 1 / 13 | |
| d | 1 | 1 / 13 | |
| r | 1 | 1 / 13 | |
| i | 1 | 1 / 13 | |
| a | 1 | 1 / 13 | |