

Kuis ke-4 IF2091 Struktur Diskrit (3 SKS) – Pohon dan Kompleksitas Algoritma
 Dosen: Rinaldi Munir & Harlili
 Rabu, 9 Desember 2008
 Waktu: 55 menit

1. Sebuah surat berantai dimulai dari seorang mengirimkan surat ke 10 orang lainnya. Setiap orang diminta untuk mengirimkan surat tersebut kepada 10 orang lainnya dan pada surat itu harus terkandung daftar nama 6 orang pengirim sebelumnya. Bila sudah terdapat 6 buah nama, maka tiap orang yang menerima surat tersebut harus mengirimkan satu dollar ke nama paling atas pada daftar tersebut. Nama tersebut dihapus dan sang penerima menambahkan namanya di dasar daftar tersebut. Bila diasumsikan semua orang mengirim surat tersebut dan tidak ada yang menerima surat lebih dari sekali, berapakah uang yang akan didapat seseorang bila mengikuti surat berantai tersebut? Pohon apakah yang terbentuk? (20)

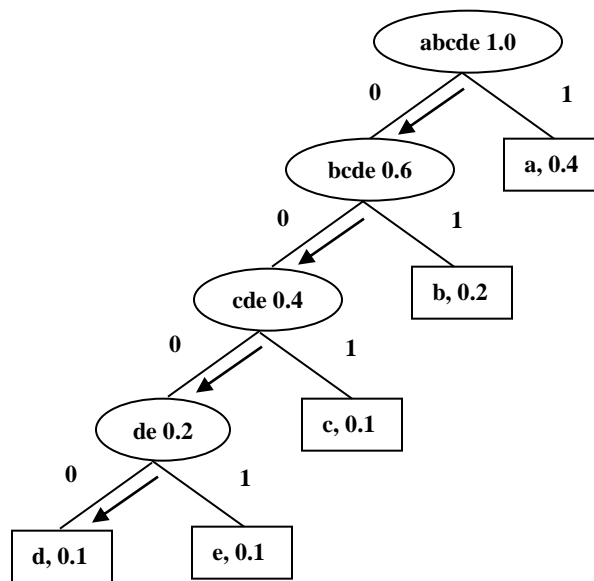
Solusi:

Anggap P adalah orang yang mengirimkan surat berantai. Maka akan ada 10 orang yang menerima surat dari P dengan keadaan P pada urutan ke-6 di daftar pengirim. Kemudian 100 orang akan menerima surat di mana nama P berada di posisi 5, begitu seterusnya hingga terbentuk pohon 10-ary penuh dengan ketinggian 6. Pohon tersebut memiliki 1.000.000 daun, sehingga P akan menerima \$ 1.000.000.000

2. Terdapat sebuah string **abaaccdeba**
- Gambarkan pohon huffman nya
 - Tentukan kode huffman untuk masing – masing karakter (tuliskan dalam bentuk tabel)
 - Tentukan kata yang terbentuk dari rangkaian bit **10010001** dengan proses decoding menggunakan kode huffman di atas (10 + 5 + 5)

Solusi:

- a. Pohon huffman yang terbentuk



- b. Tabel kode huffman yang terbentuk

simbol	Kekerapan	frekuensi	Kode Huffman
a	4	0,4	1

<i>b</i>	2	0,2	01
<i>c</i>	2	0,2	001
<i>d</i>	1	0,1	0000
<i>e</i>	1	0,1	0001

- c. Proses decoding ini dimulai dengan pembacaan satu karakter pertama. Bit yang dibaca **1** dan kita lihat bit ini berpadanan dengan simbol **a**. selanjutnya baca karakter kedua, bit **0**. Tidak terdapat kode huffman **0**, maka baca kode bit selanjutnya **0**. Tidak ada kode Huffman yang merepresentasikan rangkaian bit **00**, maka lanjutkan pembacaan lagi. Pada pembacaan selanjutnya, rangkaian bit yang terbentuk **001**, substring ini bersesuaian dengan kode Huffman yang memetakan ke simbol **c**. begitu seterusnya hingga seluruh string biner berhasil dibaca. Pada akhirnya kita akan mendapatkan **ace** dari pembacaan string biner **10010001**

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

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

Solusi:

$$\begin{aligned} T(n) &= k + k^2 T(n-1) \\ &= k + k^2 (k + k^2 T(n-2)) = k + k^3 + k^4 T(n-2) \\ &= k + k^3 + k^4 (k + k^2 T(n-3)) = k + k^3 + k^5 + k^6 T(n-3) \\ &\vdots \\ &= (k + k^3 + k^5 + \dots + k^{2n-3}) + k^{2n-2} T(1) \\ &= k + k^3 + k^5 + \dots + k^{2n-3} + k^{2n-1} \\ &= \frac{k(k^{2n} - 1)}{k^2 - 1} = O(k^{2n-1}) \end{aligned}$$

4. Berikan estimasi Big-O untuk $t(n) = (n + 5)\log(n^2 + 1) + 3n^2$ (15)
 Untuk $f(n) = n + 5 \rightarrow f(n) = O(n)$

Untuk $g(n) = \log(n^2 + 2)$

$$\begin{aligned} \log(n^2 + 1) &\leq \log(2n^2) = \log 2 + \log n^2 \\ &= \log 2 + 2 \log n \leq \log n, \text{ untuk } n > 2 \end{aligned}$$

Untuk $h(n) = 3n^2 \rightarrow h(n) = O(n^2)$

$$\begin{aligned} \text{Notasi Big-O untuk } s(n) &= f(n)g(n) + h(n) \\ &= \max(n \cdot \log n, n^2) = O(n^2) \end{aligned}$$

5. Tentukan kompleksitas waktu dari algoritma dibawah ini jika melihat banyaknya jumlah proses $a \leftarrow a + 1$

```

for i  $\leftarrow$  1 to n do
  for j  $\leftarrow$  1 to i do
    for k  $\leftarrow$  1 to j do
      a  $\leftarrow$  a + 1
    endfor
  endfor
endfor

```

Tentukan pula nilai O -besar, Ω -besar, dan Θ -besar dari algoritma diatas (harus penjelasan) (25)

Solusi:

$$T(n) = 1 + (1 + 2) + \dots + (1 + 2 + \dots + n) = 1 + 3 + \dots + n(n+1)/2 = n^3/6 + n^2/2 + n/3$$

$$T(n) \leq 1/6 \cdot (n^3 + 3n^3 + 2n^3) = n^3 = O(n^3)$$

$$T(n) = \Omega(n^3)$$

$$T(n) = \theta(n^3)$$