Solusi UAS IF2151 Matematika Diskrit
Semester Ganjil Tahun 2005/2006
Hari/Tanggal: Jumat, 23 Desember 2005
Dosen: Ir. Rinaldi Munir, M.T., Ir. Dwi Hendratmo, Ph.D, Dra. Harlili, MSc.

Waktu: 110 menit

---

1. a) Gambarkan pohon ekspresi dari notasi *prefix* berikut:

$+ - \wedge\ a\ b \wedge 2\ c\ /\ 6 - b\ a$

lalu tuliskan ekspresi *postfix* dan *infix* yang bersesuaian.

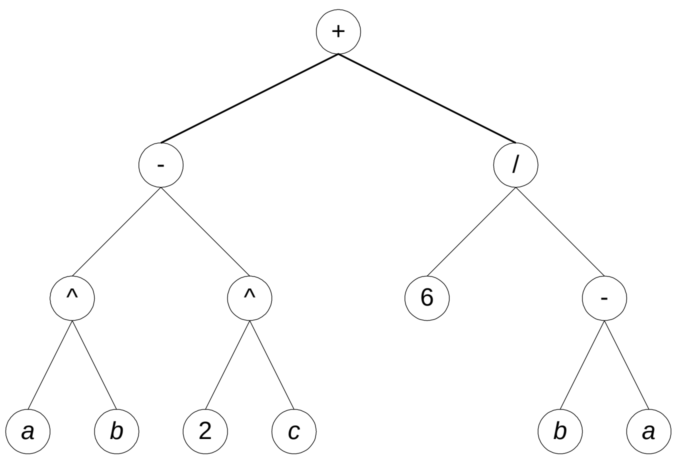
b) Berapa banyak daun pada pohon 3-*ary* dengan 100 simpul?

c) Bangunlah pohon pencarian (*search tree*) dengan urutan pembacaan data adalah *pisang*, *anggur*, *apel*, *kelapa*, *mangga*, *pepaya*, *jambu*, *rambutan*, dan durian. Berapa banyak perbandingan yang dilakukan untuk mencari kata *jeruk* di dalam pohon tersebut? (pengurutan data di dalam pohon berdasarkan huruf abjad).

(7,5 + 5 + 7,5)

Jawaban:

a)



Ekspresi *postfix* dan *infix* masing-masing diperoleh dengan melakukan traversal secara *postorder* dan *inorder*:

Postfix: $a\ b \wedge 2\ c \wedge - 6\ \ b\ a - / +$

Infix: $a \wedge b - 2 \wedge c +\ 6 / b - a$

b) Andaikan pohon tersebut pohon *m-ary* penuh. Misalkan i = jumlah simpul dalam, t = jumlah daun, maka hubungan antara jumlah simpul dalam dan jumlah daun adalah

$$(m - 1)i = t - 1$$

Jumlah simpul seluruhnya = jumlah simpul dalam + jumlah simpul daun

$$100 = t + i$$

$$i = 100 - t$$

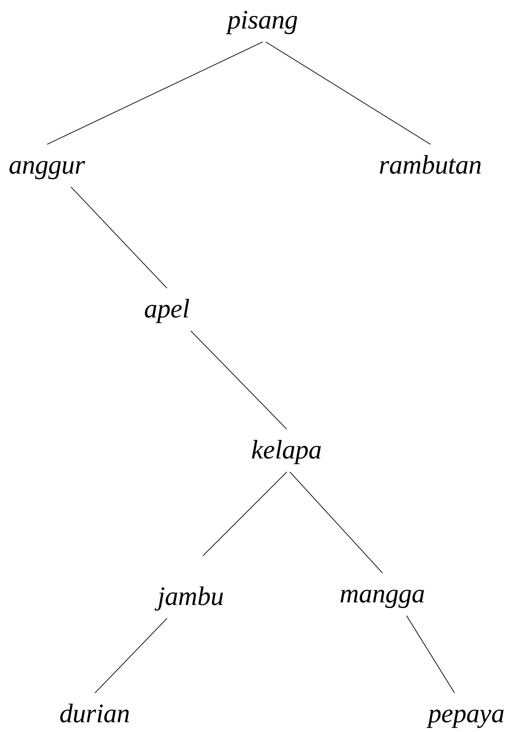
Pada pohon 3-*ary*, $m = 3$, sehingga

$$(3-1)(100-t) = t-1$$
$$200 - 2t = t - 1$$
$$201 = 3t$$
$$t = 67$$

Jadi, jumlah daun = 67

(Catatan: Jika tidak mengasumsikan pohon 3-*ary* penuh, maka hasilnya bukan bilangan bulat)

c) Pohon pencarian biner yang terbentuk:



Untuk mencari kata *jeruk* di dalam pohon di atas, pencarian dimulai dengan melakukan pembandingan mulai dari akar pohon. Kata yang dibandingkan dengan *jeruk* adalah *pisang*, *anggur*, *apel*, *kelapa*, dan *jambu*. Karena anak kanan dari simpul *jambu* adalah kosong, maka pencarian berhenti sampai di sini. Jadi hanya ada 5 kali pembandingan dengan tidak menemuka kata *jeruk*.

2. a) Diberikan sebuah *string* bit yang panjangnya n bit. Perkirakan dalam notasi O jumlah perbandingan yang diperlukan oleh algoritma untuk menentukan jumlah bit 1 di dalam rangkaian bit tersebut. Apakah jumlah perbandingan ini sama untuk kasus terbaik, kasus rata-rata, dan kasus terburuk? **(10)**

b) Nyatakan dalam notasi O kompleksitas waktu berikut :

$T(n) = (n \log n + 1)^2 + (\log n + 1)(n^2 + 1)$ dan

$T(n) = (n! + 2^n)(n^3 + \log(n^2 + 1)$ **(10)**

Jawaban:

a) Algoritma menghitung jumlah 1 mulai dari bit ke-1 sampai bit ke-n:

```
jumlah ← 0
for i ← 1 to n do
  if aᵢ = 1 then
    jumlah ← jumlah + 1
  endif
endfor
```

Jumlah operasi perbandingan adalah n kali untuk semua kasus (terbaik, rata-rata, dan terburuk), karena semua bit harus terkena pembandingan. Jadi, kompleksitas asimptotik algoritma tersebut adalah $O(n)$.

b) (i) $T(n) = (n \log n + 1)^2 + (\log n + 1)(n^2 + 1) = ((n \log n)^2 + 2n \log n + 1) + (n^2 \log n + \log n + n^2 + 1)$

$= n^2 \log^2 n + n^2 \log n + 2n \log n + \log n + n^2 + 2$

Suku yang dominan untuk n yang besar adalah $n^2 \log^2 n$, sehingan kompleksitas asimptotik ditentukan oleh suku ini. Jadi, $T(n) = O(n^2 \log^2 n) = O((n \log n)^2)$.

Cara lain: $T(n) = n^2 \log^2 n + n^2 \log n + 2n \log n + \log n + n^2 + 2$

$\leq n^2 \log^2 n + n^2 \log^2 n + 2n^2 \log^2 n + n^2 \log^2 n + n^2 \log^2 n + 2n^2 \log^2 n$

$= 8n^2 \log^2 n$ untuk $n \geq 10$

Di sini $C = 8$ dan $n_0 = 10$, sehingga menurut definisi O-besar, $T(n) = O(n^2 \log^2 n) = O((n \log n)^2)$.

(ii) $T(n) = (n! + 2^n)(n^3 + \log(n^2 + 1)) = n!n^3 + n! \log(n^2 + 1) + 2^n n^3 + 2^n \log(n^2 + 1)$
Suku yang dominan adalah $n!\, n^3$, sehingga kompleksitas asimptotik ditentukan oleh suku ini.
Jadi, $T(n) = O(n!\, n^3)$.
Cara lain: untuk $n > 3$, $n!n^3 \geq 2^n n^3$ dan $n!n^3 \geq n! \log(n^2 + 1) \geq 2^n \log(n^2 + 1)$ sehingga
$T(n) = n!n^3 + n! \log(n^2 + 1) + 2^n n^3 + 2^n \log(n^2 + 1)$
$\leq n!n^3 + n!n^3 + n!n^3 + n!n^3 = 4\, n!n^3$
Di sini $C = 4$ dan $n_0 = 3$, sehingga menurut definisi O-besar, $T(n) = O(n!n^3)$.

3. Diberikan huruf-huruf A, B, C, D, E, F. Tentukan berapa banyak rangkaian 5 huruf dimana urutan tidak diperhatikan dan boleh ada pengulangan. **(15)**

Jawaban:

Problem ini adalah kombinasi dengan pengulangan dimana $n = 6$, $r = 5$, sehingga
$C(n + r - 1, r) = C(6 + 5 - 1, 5) = C(10,5) = 252$

4. Tentukan berapa banyak bilangan bulat positif lebih kecil dari 1.000.000 yang mempunyai jumlah dari digit-digitnya sama dengan 19. **(15)**

Jawaban:
Bilangan bulat positif < 1.000.000, bilangan yang paling besarnya = 999.999, jadi ada 6 digit.
Problem ini adalah mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $0 \leq X_i \leq 9$. (Nilai : 8)

i) Mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $0 \leq X_i \leq 19$. (Nilai: 3)
Dengan menggunakan kombinasi dengan pengulangan diperoleh
$C(6+19-1,19) = C(24,19) = 42.504$

ii) Mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $X_1 \geq 10$. (Nilai : 3)
Dengan menggunakan kombinasi dengan pengulangan diperoleh
$C(6+9-1,9) = C(14,9) = 2.002$

Hal yang sama dilakukan untuk X_2, X_3, X_4, X_5, X_6, mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $X_i \geq 10$.

Dengan demikian ada 6 kali dan masing-masing mempunyai jawab
$C(6+9-1,9) = C(14,9) = 2.002$, sehingga jawab mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $X_i \geq 10$ dan i= 1,2,3,4,5,6 adalah
6 x 2.002 = 12.012

iii) Mencari solusi dari persamaan (Nilai : 1)
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $0 \leq X_i \leq 9$ sama dengan
Mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $0 \leq X_i \leq 19$ dikurangi
Mencari solusi dari persamaan
$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 19$ dimana $X_i \geq 10$ dan i= 1,2,3,4,5,6 diperoleh jawabnya =
42.504 – 12.012 = 30.492

5. *Half Adder* adalah sebuah sirkuit dengan dua *input x* dan y, serta dua *output s* dan *c*. Tabel kebenearan sirkuit ini adalah sebagai berikut:

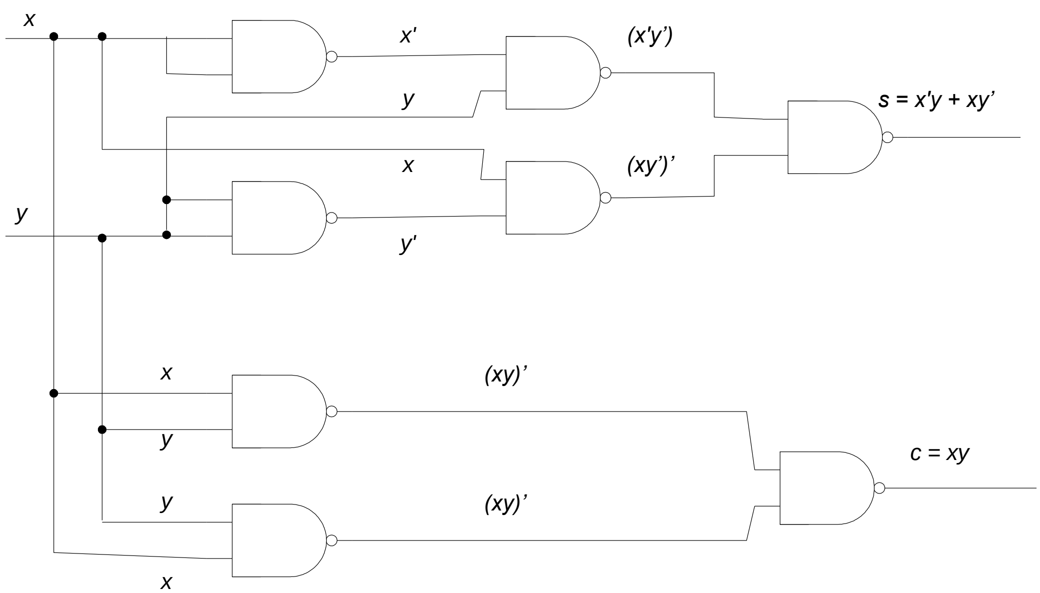
Input		**Output**	
x	*y*	*s*	*c*
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Buat rangkaian logika untuk *half adder* di atas hanya dengan menggunakan gerbang-gerbang logika *NAND* (asumsi: setiap gerbang NAND hanya memiliki dua input)!

Jawaban:

$$
\begin{aligned}
s &= xy' + x'y \\
&= ((xy' + x'y)')' \\
&= ((xy')'.(x'y)')'
\end{aligned}
$$

$$
\begin{aligned}
c &= xy \\
&= (xy + xy) \\
&= ((xy + xy)')' \\
&= ((xy)'.(xy)')'
\end{aligned}
$$



6. Untuk setiap soal di bawah, sebutkan apakah ada geraf sederhana dengan lima simpul (*vertex*) yang memiliki derajat untuk masing-masing simpul sebagai berikut? Jika ada, gambar gerafnya!

   a) 3,3,2,3,3
   b) 4,3,1,4,2
   c) 2,1,3,0,2
   d) 4,4,3,3,3

Jawaban:

a) Ada
b) Tidak Ada
c) Ada
d) Tidak Ada (jumlah derajad ganjil)