

Tugas 3 Praktikum Analgo

- 1.) Untuk $T(n) = 2 + 4 + 8 + 16 + \dots + 2n$, tentukan nilai dari C , $f(n)$, n_0 , dan notasi Big-O

Jawab

$T(n)$ = deret geometri

$$S(n) = a \frac{(r^n - 1)}{r - 1} = 2 \frac{(2^n - 1)}{2 - 1} = 2 \cdot 2^n - 2$$

$$T(n) = 2 \cdot 2^n - 2 = 2^{n+1} - 2$$

$$T(n) = O(2^n) \Rightarrow \text{notasi Big-O} // \quad f(n) = 2^n //$$

Untuk $C=2$ dan $n \geq 0$ dan $n_0=0$, dibuktikan $T(n) \leq C \cdot f(n)$

$$2^{n+1} - 2 \leq C \cdot 2^n$$

$$\frac{2^{n+1}}{2^n} - \frac{2}{2^n} \leq C$$

$$2 - \frac{2}{2^n} \leq C$$

$$C \geq 0 \text{ maka } n_0 = 1 \text{ dan } C \geq 1$$

- 2.) Buktikan untuk konstanta-konstanta positif p, q , dan r : $T(n) = pn^2 + qn + r$ adalah $O(n^2)$, $\Omega(n^2)$ dan $\Theta(n^2)$

Jawab = • $pn^2 + qn + r \leq pn^2 + qn^2 + rn^2 \rightarrow T(n) \leq C \cdot f(n) \rightarrow O(n^2)$

Jika $n=1 \rightarrow p+q+r \leq p+q+r \Rightarrow \text{terbukti} //$

• $T(n) \geq C \cdot f(n)$

$$C = p+q+r$$

$$pn^2 + qn + r \geq C \cdot f(n)$$

$$= 1+1+1 = 3 \rightarrow C \leq 3 \text{ dan } n=1$$

$$pn^2 + qn + r \geq C \cdot n^2$$

$$T(n) = pn^2 + qn + r \rightarrow \Omega(n^2) \Rightarrow \text{terbukti} //$$

$$p + \frac{q}{n} + \frac{r}{n^2} \geq C$$

• $T(n) = O(f(n))$ Jika $T(n)$ adalah $O(f(n))$ dan $\Omega(f(n))$
 $f(n) = n^2$, maka $T(n) = O(f(n)) = \Omega(f(n)) \rightarrow \Theta(n^2)$

Terbukti //

3.) Tentukan waktu kompleksitas asimptotik (Big-O, Big- Ω , Big- Θ) dari program berikut

```

for k ← 1 to n           O(n)
  for i ← 1 to n         O(n)
    for j ← 1 to n do    O(n)
      wij ← wij or wik and wkj  O(1)
    end for
  end for
end for

```

Jawaban

$$T(n) = O(n) \cdot O(n) \cdot O(n) \cdot O(1) = O(n^3) \rightarrow f(n)$$

$$\text{Big-O} = O(f(n)) = O(n^3)$$

$$\text{Big-}\Omega = \Omega(f(n)) = O(n^3)$$

$$\text{Big-}\Theta \Rightarrow \text{terpenuhi karena Big-O} = \text{Big-}\Omega$$

$$\text{Big-}\Theta = \Theta(f(n)) = \Theta(n^3)$$

4.) Tulis algoritma untuk menjumlahkan 2 matriks masing masing berukuran $n \times n$.
Berapa kompleksitas waktunya $T(n)$? Berapa kompleksitas waktunya dalam asimptotik (O, Ω , Θ)?

Jawab : Deklarasi $i, j = \text{integer}$

(1) for $i \leftarrow 1$ to n

(2) for $j \leftarrow 1$ to n do

(3) hasil $[i, j] \leftarrow a[i, j] + b[i, j]$

(4) end for

(5) endfor

$$\text{maka } T(n) = n \cdot n = n^2 \rightarrow f(n)$$

$T(n)$ sudah berlaku worst case

sehingga Big-O = Big- Ω

$$\text{Big-O} = O(f(n)) = \Omega(n^2)$$

$$\# \text{ Big-O} = \text{Big-}\Omega \text{ maka Big-}\Theta = \text{Big-O} = \text{Big-}\Omega = \Theta(f(n))$$

5. Tulislah algoritma untuk mengcopy isi sebuah linked list ke linked list lain. Untuk ukuran elemen linked list adalah n . Berapa kompleksitasnya? Berapa kompleksitas waktu asimtotik?

Jawaban :

(1) for $i \leftarrow 1$ to n $\rightarrow n$ kali ($O(n)$)

(2) $b[i-1] \leftarrow a[i-1]$

(3) endfor

maka $T(n) = n$

$T(n)$ = worst case & best case juga

sehingga Big O = Big Ω

$$T(n) = n, O(f(n)) = O(n) \quad \left. \begin{array}{l} O(f(n)) = \Theta(n) \\ \Omega(f(n)) = \Omega(n) \end{array} \right\}$$

6. Diberikan Bubble Sort

- Berapa jumlah operasi perbandingan elemen?
- Berapa kali maks pertukaran elemen tabel?
- Hitung kompleksitas asimtotik!

Jawab

jumlah operasi

Pass	Jumlah Operasi	maka
1	$n-1$	
2	$n-2$	$T(n) = (n-1) + (n-2) + \dots + 1$ $= \frac{n(n-1)}{2} = \frac{(n^2 - n)}{2}$
3	$n-3$	
\vdots	\vdots	
n	1	

maks pertukaran elemen = $\frac{n(n-1)}{2}$

Kompleksitas waktu Asimtotik

Big- $O \rightarrow T(n) = \frac{n^2 - n}{2} = O(n^2)$

Big- $\Omega \rightarrow T(n) = n^2 + n = \Omega(n^2)$

Big- $\Theta \rightarrow T(n) = n^2 + n = \Theta(n^2)$

7.) Problem x dengan ukuran $n=8$, algoritma mana yang paling cepat?
Algoritma mana yang paling cepat secara asimtotik?

- Algoritma A, $T(n) = O(\log N) = O(\log 8)$
- " B, $T(n) = O(N \log N) = O(8 \log 8)$
- " C, $T(n) = O(N^2) = O(64)$

Jawaban > algoritma A

8.) Algoritma mengevaluasi polinom yang lebih baik dapat dibuat dengan metode horner

$$p(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + a_n x)))$$

func p^2 (inp x : real) \rightarrow real

deklarasi

k : int

b_1, b_2, \dots, b_n : real

Algoritma

$b_n \leftarrow a_n$

for $k \leftarrow n-1$ down to 0 do

$b_k \leftarrow (a_k + b_{k+1}) \times x$

end for

return b_0

Hitung operasi kali & jumlahnya, jumlahkan keduanya dan tentukan KWA nya terbaik p atau p^2 ?

Jawab

- Algoritma p \rightarrow jumlah = n kali, ~~n~~ kali $T(n) = n+n = 2n = n$
- " " p^2

$$T_2(n) = 1+n = O(n)$$

- keduanya sama baiknya karena $T(n)$ & kompleksitas asimtotiknya sama sama bernilai $O(n)$