

Nama: Shalvina Zahwa Aulia
NPM: 140810180052

Kelas: B

Tugas Praktikum Analgo

No
Date

Katikan Analisa Modul Praktikum 3

- 1) Untuk $T(n) = 2 + 4 + 8 + 16 + \dots + 2^n$, tentukan nilai C , $f(n)$, n_0 , dan notasi big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C \cdot f(n)$ untuk $n \geq n_0$.

Jawab:

$$T(n) = 2 + 4 + 8 + 16 + \dots + 2^n = O(2^n)$$

$$T(n) \rightarrow \text{deret geometri} = \frac{a(r^n - 1)}{r - 1} = \frac{2(2^n - 1)}{2 - 1} = 2^{n+1} - 2$$

$$T(n) \leq C \cdot 2^n$$

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

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

$$2 - \frac{2}{2^n} \leq C \Leftrightarrow n_0 = 1, \quad 1 \leq C$$

$$\therefore f(n) = 2^n$$

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

Jawab:

① $T(n) = pn^2 + qn + r = O(n^2)$

$$T(n) \leq C \cdot f(n)$$

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

$$\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \leq C \quad \longleftrightarrow \quad \text{misal } n_0 = 1$$

$$p + q + r \leq C \quad \longleftrightarrow \quad \text{misal } p = q = r = 1$$

$$\text{maka } 3 \leq C$$

$$n_0 = 1$$

$$q \geq 1$$

② $T(n) = pn^2 + qn + r = \Omega(n^2)$

$$T(n) \geq c \cdot (g(n))$$

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

$$\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \geq c \quad \longleftrightarrow \quad \text{misal } n_0 = 1$$

$$p + q + r \geq c \quad \longleftrightarrow \quad \text{misal } p = q = r = 1$$

$$3 \geq c$$

③ $T(n) = pn^2 + qn + r = \Theta(n^2)$
 $O(n^2)$ dan $\Omega(n^2)$ benar
dan berderajat
sama, maka $\Theta(n^2)$
benar

3) Tentukan waktu kompleksitas asimtotik (Big-O, Big-Ω, dan Big Θ) dari kods program berikut :

```

for k ← 1 to n do
  for r ← 1 to n do
    for j ← 1 to n do
      wij ← wij / r or wrk and wrkj
    endfor
  endfor
endfor

```

Jawab

• Operasi assignment

$w_{ij} \leftarrow w_{ij} \text{ or } w_{ik} \text{ and } w_{kj} \rightarrow n^3$

↳ $T(n) = n^3$

↳ Big-O : $O(n^3)$

$$n^3 \leq c \cdot n^3$$

$$1 \leq c$$

↳ Big-Ω : $\Omega(n^3)$

$$n^3 \geq c \cdot n^3$$

$$1 \geq c$$

↳ Big Θ : $\Theta(n^3)$

Karena $O(n^3)$ & $\Omega(n^3)$

berderajat sama, maka

$$\Theta = n^3 \quad \Theta(n^3)$$

4) Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran $n \times n$. Berapa kompleksitas waktunya $T(n)$? Berapa kompleksitas waktu asimptotikanya yang dinyatakan dalam Big-O, Big-Ω dan Big-Θ?

Jawab :

• algoritma :

```

for i ← 1 to n do
  for j ← 1 to n do
    mij ← aij + bij
  endfor
endfor

```

$$T(n) = n^2$$

for j ← 1 to n do

$$\rightarrow O(n^2)$$

$$\rightarrow \Omega(n^2)$$

$$\rightarrow \Theta(n^2)$$

mij ← aij + bij

$$n^2 \leq c \cdot n^2$$

$$n^2 \geq c \cdot n^2$$

.....

endfor

$$1 \leq c$$

$$1 \geq c$$

endfor

5) Tulislah algoritma untuk menyalin isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktu $T(n)$? Big-O, Big-Ω dan Big-Θ?

Jawab :

```

for i ← 1 to n do
  bi ← ai
endfor

```

$$\rightarrow \text{Big-O} \quad \rightarrow \text{Big-}\Omega \quad \rightarrow \Theta(n)$$

bi ← ai

$$O(n)$$

$$\Omega(n)$$

endfor

$$n \leq c \cdot n$$

$$n \geq c \cdot n$$

↳ $T(n) = n$

$$1 \leq c$$

$$1 \geq c$$

Kelas B Latihan analisa modul 3 analgo

6) (Ada algoritma bubblesort)

a) Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!

Jawab:

Operasi perbandingan sebanyak $\frac{n(n-1)}{2}$ kali //

b) Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan!

Jwb: $\frac{n(n-1)}{2}$ kali //c) Hitung Big-O, Big- Ω , Big- Θ • Best case (Big- Ω) jika semua elemen sudah terurut

$$\rightarrow T(n) = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} \rightarrow T_{\min}(n) = \Omega(n^2)$$

$$T_{\min}(n) = \frac{n^2}{2} - \frac{n}{2} \geq c \cdot n^2$$

$$\frac{1}{2} - \frac{1}{2n} \geq c \leftarrow \text{misal } n_0 = 1$$

$$0 \geq c$$

• Worst case (Big-O)

• Perbandingan : $\frac{n(n-1)}{2}$ kali• Assignment : $3 \frac{n(n-1)}{2}$ kali

$$\left. \begin{array}{l} \text{Perbandingan} \\ \text{Assignment} \end{array} \right\} T_{\max}(n) = \frac{4n(n-1)}{2}$$

$$= 2n^2 - 2n = O(n^2)$$

$$T_{\max}(n) = 2n^2 - 2n \leq c \cdot n^2$$

$$2n^2 - 2n \leq c \cdot n^2$$

$$2 - \frac{2}{n} \leq c \leftarrow \text{misal } n_0 = 1$$

$$0 \leq c$$

• $\Theta(n^2) = T(n)$ karena $O(n^2) \leftarrow \Omega(n^2)$ benar & berderajat sama.

7) Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma ($N=8$) mana algoritma yg paling cepat? Secara asimtotik, algoritma mana yang paling cepat?

Jawab: $\log 2 = 0,3$

$$a) \text{ Algoritma A } \rightarrow T(n) = O(\log N) = O(3 \log 2) = O(0,9)$$

$$b) \text{ Algoritma B } \rightarrow T(n) = O(N \log N) = O(8 \cdot 3 \log 2) = O(7,2)$$

c) Algoritma c $\rightarrow T(n) = O(N^2) = O(64)$

\therefore Algoritma yang paling cepat adalah algoritma A //

8) Algoritma mengevaluasi polinom dg metode horner.

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + a_n x))) \dots)$$

Hitung berapa operasi perkalian & penjumlahan dr algoritmanya, jumlah kedua hitungan tsb, Big-O, manakah yg terbaik, algoritma p atau p2?

Jawab:

$T(n) \rightarrow$ assignment \rightarrow $b_n \leftarrow a_n$ $1x$
 $b_k \leftarrow a_k + b_{k+1} * x \leftarrow n x$

$$T(n) = 1 + n$$

$$\text{Big-O } p2 = O(n)$$

Algoritma p:

pertambahan : $n x$ $T(n) = 2n$

perkalian : $n x$

$$\text{Big-O } p = O(2n)$$

\therefore algoritma p2 lebih baik dari p