

Nama : Hana Meilina Fauziyyah

NPM : 140810180012

Kelas : B

1. Untuk $T(n) = 2 + 4 + 6 + 8 + 16 + \dots + n^2$, tentukan nilai C , $f(n)$, n_0 , dan notasi big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C$ untuk semua $n \geq n_0$.

Jawab :

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

Notasi Big-O $\rightarrow O(2^n)$

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

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

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

$$C \geq 1$$

2. Buktikan bahwa untuk konstanta² positif p, q, r :
 $T(n) = pn^2 + qn + r$ adalah $O(n^2)$, $\Omega(n^2)$, dan $\Theta(n^2)$

Jawab :

\rightarrow Pembuktian Big-O ($O(n^2)$)

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

$$pn^2 + qn + r \leq Cn^2$$

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

$$p + q + r \leq C, \quad \text{untuk } p=1, q=1, r=1$$

$$C \geq 3$$

\rightarrow Pembuktian Big- Ω ($\Omega(n^2)$)

$$T(n) \geq c(f(n))$$

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

$$\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \geq C, \quad n_0 = 1$$

$$C \leq p + q + r, \quad (p, q, r = 1)$$

$$C \leq 3$$

\rightarrow Pembuktian Big- Θ

Karena Big-O dan Big- Ω benar dan berderajat yg sama juga, ($O(n^2)$ dan $\Omega(n^2)$) maka Big- Θ juga sama yaitu $\Theta(n^2)$ terbukti benar.

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

```
for k ← 1 to n do
  for i ← 1 to n do
    for j ← 1 to n do
       $w_{ij} \leftarrow w_{ij}$  or  $w_{ik}$  and  $w_{kj}$ 
    end for
  end for
end for
```

Jawab :

Operasi assignment

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

$$\text{maka } T(n) = n^3$$

\rightarrow Untuk Big-O $\rightarrow O(n^3)$

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

$$n^3 \leq C \cdot n^3$$

$$C \geq 1$$

\rightarrow Untuk Big- $\Omega \rightarrow \Omega(n^3)$

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

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

$$C \leq 1$$

\rightarrow Untuk Big- $\Theta \rightarrow \Theta(n^3)$

Karena Big-O dan Big- Ω berderajat sama yaitu $O(n^3)$ dan $\Omega(n^3)$ maka Big- Θ juga sama yaitu $\Theta(n^3)$

4. Tulislah algoritma untuk menjumlahkan 2 buah matriks yang masing-masing berukuran $n \times n$. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimtotiknya yang dinyatakan dalam Big-O, Big- Ω , dan Big- Θ ?

Jawab :

→ Algoritma
 for $i \leftarrow 1$ to n do
 for $j \leftarrow 1$ to n do
 $m_{ij} \leftarrow a_{ij} + b_{ij}$
 end for
 end for
 $T(n) = n^2$

→ Big O $\rightarrow O(n^2)$
 $n^2 \leq C \cdot n^2$
 $C \geq 1$

→ Big $\Omega \rightarrow \Omega(n^2)$
 $n^2 \geq C \cdot n^2$
 $C \leq 1$

→ Big $\Theta \rightarrow \Theta(n^2)$
 Karena Big O dan Big Ω berderajat sama yaitu $O(n^2)$ dan $\Omega(n^2)$ maka Big Θ yaitu $\Theta(n^2)$

5. Tulislah Algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimptotiknya yg dinyatakan dalam Big-O, Big Ω , Big Θ ?

Jawab:

→ Algoritma
 for $i \leftarrow 1$ to n do
 $a_i \leftarrow b_i$
 end for
 $n = T(n)$

→ Big O
 $n \leq C \cdot n$
 $1 \leq C$
 $C \geq 1$

→ Big Ω
 $n \geq C \cdot n$
 $1 \geq C$
 $C \leq 1$

→ Big Θ
 $\Theta(n)$ karena Big O dan Big Ω nya sama ($O(n)$) dan berderajat sama.

6. Diberikan Algoritma Bubble Sort

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

Jawab:

$$1 + 2 + 3 + 4 + \dots + (n-1) = \frac{n(n-1)}{2} \text{ kali}$$

b. Berapa kali maksimum pertukaran elemen² tabel dilakukan?

Jawab:

$$\frac{n(n-1)}{2} \text{ kali}$$

c. Hitung kompleksitas waktu asimptotik (Big-O, Big- Ω , dan Big- Θ) dr algoritma Bubble Sort tersebut!

Jawab:

→ Best Case

$$\frac{(n-1)n}{2} \text{ kali}$$

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

→ Worst Case

$$\text{Perbandingan} \rightarrow \frac{n(n-1)}{2}$$

$$\text{Memasukkan nilai} \rightarrow \frac{3n(n-1)}{2}$$

$$T_{\max}(n) = \frac{4n(n-1)}{2} = 2n^2 - 2n$$

→ Big-O

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

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

$$2 - 2 \leq C$$

$$C \geq 0$$

→ Big- Ω

$$\frac{n^2 - n}{2} \geq C \cdot n^2$$

$$\frac{1}{2} - \frac{1}{2n} \geq C, \quad n_0 = 1$$

$$\frac{1}{2} - \frac{1}{2} \geq C$$

$$C \leq 0$$

→ Big Θ

n^2 , karena Big O dan Big Ω berderajat sama.

7. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:

- a. Algoritma A mempunyai kompleksitas waktu $O(\log N)$
- b. Algoritma B _____ $O(N \log N)$
- c. Algoritma C _____ $O(N^2)$

Untuk problem X dgn ukuran $N=8$, algoritma manakah yg paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

Jawab:

- a. $O(\log 8) = O(3 \log 2)$
- b. $O(8 \log 8) = O(24 \log 2)$
- c. $O(8^2) = O(64)$

Jadi yg paling efektif adalah algoritma A karena semakin kecil maka semakin efektif.

8. Operasi memasukan nilai

- $b_n \leftarrow a_n$ 1 kali
- $b_k \leftarrow a_k + b_k + 1 + x$ n kali

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

$$O(n) = \text{untuk } p^2$$

Algoritma P

Pengjumlahan n kali

Pertalian n kali

$$T(n) = 2n$$

maka p^2 lebih baik dari P