

Nama : Shamarhena Chairani

NPM : 140810180022

Kelas : B

Jawaban Modul Praktikum 3

1. $T(n) = 2 + 4 + 6 + 8 + 10 + \dots + 2^n$

Tentukan nilai C , $f(n)$, n_0 , dan notasi Big O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C$ untuk semua $n \geq n_0$.

Jawab :

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

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

$$\begin{array}{l|l} T(n) \leq C \cdot 2^n & 2 - \frac{2}{2^n} \leq C \\ 2^{n+1} - 2 \leq C \cdot 2^n & 2 - 1 \leq C \\ \frac{2^{n+1}}{2^n} - \frac{2}{2^n} \leq C & 1 \leq C \\ 2 - \frac{2}{2^n} \leq C, n_0 = 1 & C \geq 1 \end{array}$$

2. Buktikan bahwa 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 :

→ Pembuktian Big-O ($O(n^2)$)

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

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

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

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

$$C \geq p + q + r \rightarrow C \geq 1 + 1 + 1 \rightarrow C \geq 3$$

→ Pembuktian big- Ω ($\Omega(n^2)$)

$$T(n) \geq C \cdot F(n)$$

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

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

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

$$C \leq 1 + 1 + 1 \rightarrow C \leq 3$$

→ Pembuktian big- Θ

Karena big-O dan big- Ω benar dan berderajat yang sama juga yaitu $O(n^2)$ 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[i] ← w[i] or w[k] and w[j]
    end for
  end for
end for
```

Jawab :

Kompleksitas waktu

Operasi Assignment

$w[i] \leftarrow w[i] \text{ or } w[k] \text{ and } w[j] = n^3$

maka : $T(n) = n^3$

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

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

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

$$C \geq 1$$

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

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

$$n^3 \geq C \cdot n^3$$

$$C \leq 1$$

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. Tuliskan Algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran $n \times n$.

Berapa kompleksitas waktunya $T(n)$? dan Berapa kompleksitas waktu asimtotik yang dinyatakan dalam big-O, big- Ω dan big- Θ ?

Jawab :

→ Algoritma Penjumlahan dua matriks

```
for i ← 1 to n do
```

```
  for j ← 1 to n do
```

```
    m[i] ← a[i] + b[j]
```

```
  end for
```

```
end for
```

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

→ Untuk big-O $\rightarrow O(n^2)$

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

$$C \geq 1$$

→ Untuk big- $\Omega \rightarrow \Omega(n^2)$

$$n^2 \geq C \cdot n^2$$

$$C \leq 1$$

∴ Untuk 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 juga sama yaitu $\Theta(n^2)$.

5. Tulistilah Algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen.
 Berapa kompleksitas waktu $T(n)$?
 Berapa kompleksitas asimptotiknya yang dinyatakan dalam big-O, big- Ω , dan big Θ ?
 Jawab:

Algoritma menyalin larik

```
for i ← 1 to n do
  a[i] ← b[i] → n = T(n)
endfor
```

→ Untuk big-O $n \leq C \cdot n$
 $1 \leq C$
 $C \geq 1$

→ Untuk big- Ω
 $n \geq C \cdot n$
 $1 \geq C$
 $C \leq 1$

∴ Karena big-O dan big Ω samayaitu $O(n)$ dan $\Omega(n)$ berderajat sama, maka Big Θ nya sama yaitu $\Theta(n)$

6. Diberikan algoritma bubble sort

a. Hitung berapa jumlah operasi perbandingan elemen² tabel!

Jawab:

Jumlah operasi perbandingan

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

b. Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?

Jawab:

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

c. Hitung kompleksitas waktu asimptotik!

→ Best Case (semua telah terurut)

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

→ Worst Case (semua data harus ditukar)

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

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

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

• Big O

$$2n^2 - 2n \leq Cn^2$$

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

$$2 - 2 \leq C$$

$$C \geq 0 //$$

• Big Ω

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

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

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

$$C \leq 0 //$$

∴ Karena big-O dan big Ω berderajat sama, maka big Θ sama $n^2 //$

7. Untuk menjelaskan problem X dgn ukuran N terdapat 3 macam algoritma

a). Algoritma A $\rightarrow O(\log N)$

b). Algoritma B $\rightarrow O(N \log N)$

c). Algoritma C $\rightarrow O(N^2)$

Jika $N = 8$ maka Algoritma yg mana yg paling efektif?

a). $O(\log 8) = O(\log 2^3) = O(3 \log 2)$

b). $O(8 \log 8) = O(24 \log 2)$

c). $O(8^2) = O(64)$

∴ Maka yg paling efektif adalah Algoritma A, karena semakin kecil $O()$ semakin Efektif.

8. Hitung berapa operasi perkalian dan penjumlahan yang dilakukan oleh Algoritma di soal, jumlahkan kedua hitungan tsb. Lalu tentukan kompleksitas waktu asimptotik (Big-O nya). Manakah yg terbaik algoritma Patau B?

Jawab:

Operasi Memasukkan 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

Penjumlahan n kali

Pertalian n kali

$$T(n) = 2n$$

→ maka p^2 lebih baik dari p //