Bungo Azizha N
140810180016
Kelas B

Tugas 3 Analgo

1) Untuk $T(n) = 2+4+8+16+ \dots + 2^n$, tentukan nilai c, $f(n)$, n_0 & notasi Big-O sehingga $T(n) = O(f(n))$ jika $T(n) \le c$ untuk semua $n \ge n_0$

Jawab : $T(n) \rightarrow$ deret geometri

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

$$T_n = 2 \cdot 2^n - 2 = 2^{n+1} - 2$$

$\Rightarrow T(n) = O(2^n) \Rightarrow$ notasi Big-O dan $f(n) = 2^n$

$\Rightarrow$ untuk $c = 2$ dan $n \ge 0$ dan $n_0 = 0$ dapat dibuktikan $T(n) \le c\,f(n)$

$$\frac{2^{n+1} - 2 \le c \cdot 2^n}{2 - \frac{2}{2^n} \le c} : 2^n$$

$c > 0$

maka $n_0 = 1$ dan $c \ge 1$

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

Jawab : $\Rightarrow pn^2 + qn + r \le pn^2 + qn^2 + rn^2 \rightarrow T(n) \le c \cdot f(n) \rightarrow O(n^2)$ Terbukti ✓

jika $n = 1 \rightarrow p + q + r \le p + q + r$ Terbukti ✓

$\Rightarrow T(n) \ge c \cdot f(n)$ $\quad c = p + q + r = 1 + 1 + 1 = 3 \rightarrow c \le 3$ dan $n = 1$

$pn^2 + qn + r \ge c \cdot f(n)$ $\quad T(n) = pn^2 + qn + r \rightarrow \Omega(n^2)$ Terbukti ✓

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

$p + \frac{q}{n} + \frac{r}{n^2} \ge c$

$\Rightarrow T(n) = \Theta(f(n))$ jika $T(n) = O(f(n))$ 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 asimtotik (Big-O, Big-Ω, dan Big-Θ)

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

Big-O $= O(f(n)) = O(n^3)$

Big-Ω $= \Omega(f(n)) = \Omega(n^3)$

Big-Θ $\Rightarrow$ terpenuhi karena Big-O = Big-Ω, jadi Big-Θ $= \Theta(f(n)) = \Theta(n^3)$

4). Tulis algoritma untuk menjumlahkan 2 matriks masing² berukuran $n \times n$. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimtotiknya yg dinyatakan dalam Big-O, Big-Ω & Big Θ?

Jawab : Deklarasi i, j = integer

\# Algoritma $\rightarrow$
```
(1) for i <- 1 to n do
(2)     for j <- 1 to n do
(3)         hasil [i,j] <- a[i,j]+b [i,j]
(4)     endfor
(5) endfor
```

Maka $T(n) = n \cdot n = n^2 \rightarrow f(n)$

$T(n) \rightarrow$ berlaku untuk worst case, sehingga Big-O = Big-Ω

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

Jadi,

Big-O = Big-Ω $= \Theta((f(n)))$

5) Tulis algoritma mencopy isi larik ke larik lain. untuk ukuran elemen larik = n
Berapa $T(n)$? Berapa $T(n)$ asimptotiknya?

Jawab : Algoritma $\rightarrow$
```
(1) for i <- 1 to n do
(2)     b [i-1] <- a[i-1]        -> n kali / O(n)
(3) endfor
```

maka $T(n) = n$

$T(n)$ = worst case & best case sehingga Big-O = Big-Ω

$T(n) = n$, $O(f(n)) = O(n)$
$\Omega(f(n)) = \Omega(n)$
} $O(f(n)) = \Theta(n)$

6). Diberikan Bubble sort

a. Berapa jumlah operasi perbandingan elemen²ˣ ?

b. Berapa kali maks pertukaran elemen table ?

c. Hitung kompleksitas asimptotnya !

Jawab :

a. Jumlah operasi perbandingan

Pass	Jml. operasi
1	$n-1$
2	$n-2$
3	$n-3$
⋮	
n	1

maka :

$T(n) = (n-1) + (n-2) + \ldots + 1$

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

$= \left(\frac{n^2 - n}{2}\right)$

b. Maks pertukaran elemen $= \frac{n(n-1)}{2}$

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

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

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

7). Problem x dengan ukuran $n = 8$, algoritma mana yg paling cepat? juga asimptotik?

Jawab : Semakin kecil bilangan di dalam kurung, Semakin sedikit operasi yg dikerjakan.

8). Hitung operasi kali & jumlahnya jumlahkan ke 2 nya & tentukan kompleksitas waktu asimptotnya mana yg terbaik P atau P_2 ?

Jawab : •> algoritma P $\rightarrow$ jumlah = n kali

kali = n kali

$T(n) = n + n = 2n = n$

•> Algoritma P_2 $\rightarrow$ $T_2(n) = 1 + n = O(n)$

•> Jadi, keduanya sama²ˣ baik karena $T(n)$ dan kompleksitas waktu asimptotiknya sama²ˣ bernilai $O(n)$.