Nama : Delanika Olympiani T.C
NPM : 140810180026
Kelas : B

PRAKTIKUM ANALGO
Worksheet 3.

① Dik :
$T(n) = 2+4+6+8+16+ \dots + n^2$.

Dit :
$C, f(n)$, Big O ?

Jawab :
Dapat dibuat deret geometri :

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

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

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

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

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

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

$C \geq 1$

② Dik :
p, q, r
$T(n) = pn^2 + qn + r$

Dit :
Buktikan konstanta positif p, q, r adalah $O(n^2)$, $\Omega(n^2)$, dan $\theta(n^2)$

Jawab :
- untuk Big O = $O(n^2)$

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

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

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

$p + q + r \leq C$, mis $p = q = r = 1$

$C \geq 3$

- untuk Big $\Omega = \Omega(n^2)$

$T(n) \geq C(g(n))$

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

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

$C \leq p + q + r$, mis $p = q = r = 1$

$C \leq 3$

- untuk Big - θ

$O(n^2)$ & $\Omega(n^2)$ berderajat sama dan terbukti benar maka $\theta(n^2)$ benar.

③ Dik :
Kode program

```
for k ← 1 to n do
   for i ← 1 to n do
      for j ← to n do
         Wij ← Wij or Wik and Wkj
      endfor
   endfor
endfor
```

Dit :
kompleksitas waktu asimptotik?

Jawab :
$W_{ij} \leftarrow W_{ij}$ or W_{ik} and W_{kj} dilakukan sebanyak $n \cdot n \cdot n = n^3$

$T(n) = n^3$

- untuk Big O

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

$n^3 \leq C(n^3)$

$1 \leq C$

- untuk Big Ω

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

$n^3 \geq C(n^3)$

$1 \geq C$

- untuk Big θ

Big O sama dengan Big Ω. Maka Big $\theta = n^3$ (sama)

④ Dit :
- algoritma menjumlahkan dua buah matriks? $(n \times n)$
- $T(n)$?
- Big O, Big Ω, Big θ?

Jawab :
a. algoritma :

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

b. $T(n) = n^2$
$m_{ij} \leftarrow a_{ij} + b_{ij}$ dilakukan sebanyak $n \cdot n$ kali $= n^2$.
terdapat dua looping.
$T(n) = n^2$

c. • Big O
$T(n) \leq f(n)$
$n^2 \leq C(n^2)$
$1 \leq C$

• Big Ω
$T(n) \geq f(n)$
$n^2 \geq C(n^2)$
$1 \geq C$

• Big θ
Big O = Big Ω
Maka, Big θ juga $\theta(n^2)$

⑤ Dik :
- ukuran elemen = n larik

Dit :
- algoritma menyalin isi sebuah larik ke larik lain sebanyak n elemen?
- $T(n)$?
- Big O, Big Ω, Big θ?

Jawab :
a. algoritma

```
for i ← 1 to n do
   i ← i
endfor
```

b. $T(n) = n$
karena dilakukan dalam sekali looping.

c. • Big O
$T(n) \leq f(n)$
$n \leq C(n)$
$1 \leq C$

• Big Ω
$T(n) \geq f(n)$
$n \geq C(n)$
$1 \geq C$

• Big θ
Big O = Big Ω. Maka Big $\theta = n$

⑥ Dik:
algoritma bubble sort:

```
for pass ← 1 to n - 1 do
    for k ← n down to pass + 1 do
        If ak < ak-1 then
            temp ← ak
            ak ← ak-1
            ak-1 ← temp
        endif
    endfor
endfor
```

Dit:

a. jumlah operasi perbandingan elemen-elemen tabel?

b. maks. pertukaran elemen-elemen tabel dilakukan?

c. Big O, Big Ω, Big θ?

Jawab

a. $\frac{n(n-1)}{2}$ ⇒ $(n-1)$ karena pass ← 1 dilakukan sebanyak jumlah n. dibagi 2 karena membandingkan 2 elemen

maks pertukaran elemen

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

c. Best case (semua data telah terurut)

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

Worst case (semua data melakukan pertukaran tempat)

- Perbandingan : $\frac{n(n-1)}{2}$
- assignment : $\frac{3n(n-1)}{2}$

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

① Big O

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

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

$2 - 2 \leq C$

$0 \leq C$

② Big Ω

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

$\left(\frac{\cancel{n^2}}{2}\right)\frac{1}{\cancel{n^2}} - \left(\frac{\cancel{n}}{2}\right)\frac{1}{\cancel{n^2}} \geq C.n^2$

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

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

$0 \geq C$

③ Big θ

Big O = Big Ω

maka Big θ = $\theta(n^2)$ //

⑦ Dik:

3 macam algoritma

- algoritma A : $O(\log N)$
- algoritma B : $O(N \log N)$
- algoritma C : $O(N^2)$

ukuran : N = 8

Dit:

algoritma manakah paling cepat secara asimtotik?

Jawab:

a. algoritma A : $O(\log 8)$

$O(\log 2^3) : O({}^3\log 2)$

$= O(0.631)$

b. algoritma B : $O(8 \log 8)$

$O(8 . {}^3\log 2) = O(5.048)$

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

* ${}^3\log 2 = 0.6309 \simeq 0.631$

maka yang paling cepat adalah algoritma A //

⑧ Dik:

algoritma metode Horner:

```
bn ← an
for k ← n-1 downto 0 do
    bk ← a + bk-1 * x        O(n) + O(n)
endfor
return b0
```

Dit:

- operasi perkalian & penjumlahan?
- jumlahnya?
- Big O?
- mana yang lebih baik, P / P2?

Jawab:

a. penjumlahan : n kali } P | : 1 kali } P2

perkalian : n kali + | : n kali +

b. Jumlah : 2n kali | 1+n kali

c. Big O

P = O(2n) = O(n)

P2 : O(n)

Keduanya sama bila konstanta diabaikan //