Nama : Rahma Batari
NPM : 140810180051
Kelas : A

Tugas Modul 3 - Analgo

1) $T(n) = 2+4+8+16+\ldots+2^n$

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

$T(n) = 2^{n+1}-2 = O(2^n)$

$T(n) \leqslant Cf(n)$	$n_0 = 1$
$2^{n+1}-2 \leqslant C \cdot 2^n$	$2 - \frac{2}{2} \leqslant C$
$2 - \frac{2}{2^n} \leqslant C$	$C \geqslant 1$

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

$\cdot\!> O(n^2) \rightarrow$ Big O	$n_0 = 1$
$T(n) \leqslant Cf(n)$	$p+q+r \leqslant C$
$pn^2+qn+r \leqslant Cn^2$	$C \geqslant p+q+r$
$p + \frac{q}{n} + \frac{r}{n^2} \leqslant C$	

$\cdot\!> \Omega(n^2) \rightarrow$ Big Ω	$n_0 = 1$
$T(n) \geqslant Cf(n)$	$p+q+r \geqslant C$
$pn^2+qn+r \geqslant Cn$	$C \leqslant p+q+r$
$pn + q + \frac{r}{n} \geqslant C$	

$\therefore$ Karena big O = big $\Omega = n^2$,
maka big $\theta = n^2$

3)
```
for k ← i to n do
    for i ← 1 to n do
        for j ← to n do
            w_ij ← w_ij or w_ik or w_kj ⇒ n.n.n → T(n).n^3
        endfor
    endfor
endfor
```

* big O	* Big Ω	* Big θ
$n^3 \leqslant C.n^3$	$n^3 \geqslant C.n^3$	Big O = Big Ω
$1 \leqslant C$	$C \leqslant 1$	maka Big θ pun sama
$C \geqslant 1$		Big θ = $\theta(n^3)$

4) Algoritma penjumlahan matriks $n \times m$
```
for i ← 1 to n do
    for j ← i to n do
        m_ij ← a_ij + b_ij → n.n → T(n) = n^2
    endfor
endfor
```

		* Big O = Big Ω
* Big O	* Big Ω	maka Big θ bernilai sama
$n^2 \leqslant Cn^3$	$n^2 \geqslant Cn^2$	Big $\theta = \theta(n^2)$
$1 \leqslant C$	$1 \geqslant C$	
$C \geqslant 1$	$C \leqslant 1$	

5) Algoritma menyalin larik
```
for i ← 1 to n do
    a_i ← b_i          ⇒ n = T(n)
endfor
```

* Big O	* Big Ω	* Big O = Big Ω, maka
$n \leqslant Cn$	$n \geqslant Cn$	Big θ bernilai sama
$1 \leqslant C$	$1 \geqslant C$	Big $\theta = \theta(n)$
$C \geqslant 1$	$C \leqslant 1$	

6.) a. jumlah operasi perbandingan

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

b. Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan

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

c. Hitung Kompleksitas

- Best case (semua telah terurut)

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

- Worst case (semua data harus ditukar)

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

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

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

Big O	Big Ω
$2n^2-2n \leqslant Cn^2$	$\frac{n^2-n}{2} \geqslant Cn^2$
$2 - \frac{2}{n} \leqslant C$	$1/2 - 1/2n \geqslant C$
$n_0 = 1 \rightarrow 2-2 \leqslant C$	$n_0 = 1 \rightarrow \frac{1}{2} - \frac{1}{2} \geqslant C \quad C \leqslant 0$
$C \geqslant 0$	

7) 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 mana algoritma yang paling efektif ?

a. $O(\log 8) = O({}^3\log 2) \rightarrow$ yang paling efektif

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

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

Karena makin kecil O( ) makin efektif

8.) Operasi memasukkan nilai

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

$T(n) = n+1$

$O(n)$ = untuk p^2

Algoritma P

Penjumlahan n kali

Perkalian n kali

$T(n) = 2n$

P2 lebih baik dari P