Tyko Zidane B.
190810180031
Kelas A.

Latihan Analisis Algoritma

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

bentuk

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

notasi Big O → $O(2^n)$

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

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

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

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

$C \geq 1$

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

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

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

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

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

$p + q + r \leq C$

$C \geq 3$

- Pembuktian Big-Ω ($\Omega(n^2)$)

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

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

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

$p + q + r \geq C$, misal $p = 1, q = 1, r = 1$

$C \leq 3$

- Pembuktian Big θ

karena $O(n^2)$ dan $\Omega(n^2)$ benar dan se derajat sama maka $\theta(n^2)$ terbukti benar

3. Kompleksitas waktu

operasi assignment

$w_j \leftarrow w_j$ or w_k and $w_y \rightarrow n^3$

$T(n) = n^3$

- Big O $\rightarrow O(n^3)$
  $n^3 \leq C \cdot n^3$
  $C \geq 1$

- Big - $\Omega \rightarrow \Omega(n^3)$
  $n^3 \geq C \cdot n^3$
  $C \leq 1$

- Big - $\Theta \rightarrow \Theta(n^3)$
  karena $O(n^3)$ dan $\Omega(n^3)$ berderajat sama maka $\Theta(n^3)$

4. Algoritma ~~apa~~ menjumlahkan dua matriks

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

$T(n) = n^2$

- $O(n^2)$
  $n^2 \leq C n^2$
  $C \geq 1$

- $\Omega(n^2)$
  $n^2 \geq C \cdot n^2$
  $C \leq 1$

- $\Theta(n^2)$
  $O(n^2)$ dan $\Omega(n^2)$ berderajat sama maka $\Theta(n^2)$

5. Algoritma menyalin larik

```
for i ← 1 to n do
    ai ← bi   => n = T(n)
endfor
```

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

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

- Big Θ
  $O(n)$ dan $\Omega(n)$ maka $\Theta(n)$

6. a. Jumlah operasi perbandingan

$0+1+2+3+4+\ldots+(n-1)$ x

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

b. Berapa kali pertukaran eleme-elemen tabel dilakukan (maksimum)!

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

6. c. Hitung kompleksitas waktu

Best Case (semua data sudah terurut)

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

worst case (semua data harus ditukar)

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

Assigment -> $\frac{3n(n-1)}{2}$ kali

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

- $O(n^2)$

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

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

  $C \geq 0$

- $\Omega(n^2)$

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

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

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

  $C \leq 0$

- $\theta(n^2)$

  $O(n^2)$ dan $\Omega(n^2)$ berderajat sama maka $\theta(n^2)$

7. a. algoritma A -> $O(\log N)$
   b. algoritma B -> $O(N \log N)$
   c. algoritma C -> $O(N^2)$

   $N = 8$, maka

   algoritma A -> $O(\log 8) = O(3.\log 2)$

   algoritma B -> $O(8 \log 8) = O(24.\log 2)$

   algoritma C -> $O(8^2) = O(64$

   Dengan asumsi $\log 2 = 0,301$ maka algoritma A lebih cepat daripada yang lainnya

8. Operasi assigment

   - $b_n \leftarrow a_n$ = 1 kali
   - $b_n \leftarrow a_n + b_n + x$ = n kali

   $T(n) = n + 1$

   $O(n)$ untuk P^2

   <u>Algoritma P</u>

   Pertambahan : n kali

   Perkalian = n kali

   $T(n) = 2n$

   maka algoritma P^2 lebih baik dari pada P