Nama : Firmansyah Yanuar
NPM : 1408101730051

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 → $O(2^n)$

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

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

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

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

$C \geq 1$

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 ($O(n^2)$)

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

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

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

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

$C \geq 3$

- Pembuktian Big-Ω ($\Omega(n^2)$)

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

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

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

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

$C \leq 3$

- Pembuktian Big-Θ

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

3. Kompleksitas waktu

Operasi assignment

$w_{ij} \leftarrow w_{ij}$ or w_{ik} and w_{kj} → n^3

$T(n) \geq n^3$

- Big-O → $O(n^3)$

$n^3 \leq C.n^3$

$C \geq 1$

- Big-Ω → $\Omega(n^3)$

$n^3 \geq C.n^3$

$C \leq 1$

- Big-Θ → $\Theta(n^3)$

Karena $O(n^3)$ dan $\Omega(n^3)$ berderajat sama maka $\Theta(n^3)$

4. Algoritma menjumlahkan dua matriks

```
For i ← 1 to n do
    For j ← 1 to n do
        m_ij ← a_ij + b_ij
    end for
end for
```

$T(n) = n^2$

- $O(n^2)$

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

$C \geq 1$

- $\Omega(n^2)$

$n^2 \geq C.n^2$

$C \leq 1$

- $\Theta(n^2)$

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

for i ← 1 to n do
    $a_i \leftarrow b_i$
endfor

$T(n) = n$

- $O(n)$ - $n \leq C \cdot n$, $C \geq 1$
- $\Omega(n)$ $n \geq C \cdot n$, $C \leq 1$
- $\Theta(n)$ $O(n)$ dan $\Omega(n)$ berderajat sama

6. a. Jumlah operasi perbandingan

~~Best case = 1x~~

~~Worst case~~ $1 + 2 + 3 + 4 + \dots + (n-1)x$

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

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

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

c. Hitung kompleksitas waktu

~~$T(n)$~~ 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 diurut)

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

Assignment → $3\frac{n(n-1)}{2}$ kali

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

- $O(n^2)$
  $2n^2 - 2n \leq C \cdot n^2$
  $\frac{2 - \frac{2}{n}}{} \leq C$, $n_0 = 1$
  $C \geq 2 - 2$
  $C \geq 0$
- $\Omega(n^2)$
  $\frac{n^2}{2} - \frac{n}{2} \geq C \cdot n^2$
  $\frac{1}{2} - \frac{1}{2n} \geq C$, $n_0 = 1$
  $\frac{1}{2} - \frac{1}{2} \geq C$
  $C \leq 0$
- $\Theta(n^2)$
  $O(n^2)$ dan $\Omega(n^2)$ berderajat sama

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

   $N = 8$, maka
   algoritma A → $O(\log 8) = O(3 \cdot \log 2)$
   algoritma B → $O(8 \log 8) = O(24 \cdot \log 2)$
   algoritma C → $O(8^2) = O(64)$

(bagian terpotong)
Dengan asumsi log 2 = 0.301, maka algoritma A lebih cepat daripada yang lainnya

8. Operasi assignment
   - $b_n \leftarrow a_n$: 1 kali
   - $b_k \leftarrow a_k + b_{k+1} * x$: n kali

$T(n) = n+1$

$O(n)$. untuk p2.

Algoritma P

Pertambahan : n kali
Perkalian : n kali

$T(n) = 2n$

Maka algoritma p2 lebih baik daripada p