

Tugas Praktikum Analisis Algoritma



Disusun oleh:
Rahma Batari
140810180051

Program Studi S1 Teknik Informatika
Fakultas Matematika & Ilmu Pengetahuan Alam
Universitas Padjadjaran

Latihan Analisa

1. Untuk $T(n) = 2 + 4 + 6 + 8 + 16 + \dots + n^2$, tentukan nilai C , $f(n)$, n_0 , dan notasi Big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C$ untuk semua $n \geq n_0$
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 $\Theta(n^2)$
3. Tentukan waktu kompleksitas asimptotik (Big-O, Big- Ω , dan Big- Θ) dari kode program berikut:

```
for k ← 1 to n do
  for i ← 1 to n do
    for j ← to n do
       $w_{ij} \leftarrow w_{ij}$  or  $w_{ik}$  and  $w_{kj}$ 
    endfor
  endfor
endfor
```
4. Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran $n \times n$. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big- Ω , dan Big- Θ ?
5. Tulislah algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big- Ω , dan Big- Θ ?
6. Diberikan algoritma Bubble Sort sebagai berikut:

```
procedure BubbleSort(input/output  $a_1, a_2, \dots, a_n$ , integer)
{ Mengurut tabel integer TabInt[1..n] dengan metode pengurutan bubble-
sort
Masukan:  $a_1, a_2, \dots, a_n$ 
Keluaran:  $a_1, a_2, \dots, a_n$  (terurut menaik)
}
Deklarasi
k : integer ( indeks untuk traversal tabel )
pass : integer ( tahapan pengurutan )
temp : integer ( peubah bantu untuk pertukaran elemen tabel )
Algoritma
  for pass ← 1 to n - 1 do
    for k ← n downto pass + 1 do
      if  $a_k < a_{k-1}$  then
        { pertukarkan  $a_k$  dengan  $a_{k-1}$  }
        temp ←  $a_k$ 
         $a_k \leftarrow a_{k-1}$ 
         $a_{k-1} \leftarrow temp$ 
      endif
    endfor
  endfor
```

- (a) Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
 - (b) Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
 - (c) Hitung kompleksitas waktu asimptotik (Big-O, Big- Ω , dan Big- Θ) dari algoritma Bubble Sort tersebut!
7. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:
 - (a) Algoritma A mempunyai kompleksitas waktu $O(\log N)$
 - (b) Algoritma B mempunyai kompleksitas waktu $O(N \log N)$
 - (c) Algoritma C mempunyai kompleksitas waktu $O(N^2)$Untuk problem X dengan ukuran $N=8$, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

8. Algoritma mengevaluasi polinom yang lebih baik dapat dibuat dengan metode Horner berikut:

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + a_n x)))) \dots)$$

function p2(input x : real) → real
(Mengembalikan nilai $p(x)$ dengan metode Horner)

Deklarasi

k : integer
 b_1, b_2, \dots, b_n : real

Algoritma

$b_n \leftarrow a_n$
for k ← n - 1 downto 0 do
 $b_k \leftarrow a_k + b_{k+1} * x$
endfor
return b_0

Hitunglah berapa operasi perkalian dan penjumlahan yang dilakukan oleh algoritma diatas, Jumlahkan kedua hitungan tersebut, lalu tentukan kompleksitas waktu asimptotik (Big-O)nya. Manakah yang terbaik, algoritma p atau p2?

Jawaban

Nama : Rahma Batari
NPM : 140810180051
Kelas : A

Tugas Modul 3 - Analgo

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

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

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

$$T(n) \leq C f(n) \quad \left\{ \begin{array}{l} n_0 = 1 \\ 2 - \frac{2}{2} \leq C \\ 2 - \frac{2}{2^n} \leq C \end{array} \right. \quad \left\{ \begin{array}{l} C > 1 \end{array} \right.$$

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

$$\rightarrow O(n^2) \rightarrow \text{Big } O$$

$$\left\{ \begin{array}{l} T(n) \leq C f(n) \\ pn^2 + qn + r \leq C n^2 \\ p + \frac{q}{n} + \frac{r}{n^2} \leq C \end{array} \right. \quad \left\{ \begin{array}{l} n_0 = 1 \\ p + q + r \leq C \\ C > p + q + r \end{array} \right.$$

$$\rightarrow \Omega(n^2) \rightarrow \text{Big } \Omega$$

$$\left\{ \begin{array}{l} T(n) \geq C f(n) \\ pn^2 + qn + r \geq C n^2 \\ pn + q + \frac{r}{n} \geq C \end{array} \right. \quad \left\{ \begin{array}{l} n_0 = 1 \\ p + q + r \geq C \\ C \leq p + q + r \end{array} \right.$$

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

$$3) \text{ for } k \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \text{for } i \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \quad \text{for } j \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \quad \quad w_{ij} \leftarrow w_{ij} \text{ or } w_{ik} \text{ or } w_{kj} \rightarrow n \cdot n \cdot n \rightarrow T(n) \cdot n^3$$

$$\quad \quad \quad \text{endfor}$$

$$\quad \quad \text{endfor}$$

$$\quad \text{endfor}$$

$$\begin{array}{lll} * \text{ Big } O & * \text{ Big } \Omega & * \text{ Big } \Theta \\ n^3 \leq C n^3 & n^3 \geq C n^3 & \text{Big } O = \text{Big } \Omega \\ 1 \leq C & C \leq 1 & \text{maka Big } \Theta \text{ pun sama} \\ C \geq 1 & C \leq 1 & \text{Big } \Theta = \Theta(n^3) \end{array}$$

$$4) \text{ Algoritma penjumlahan matriks } n \times n$$

$$\text{for } i \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \text{for } j \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \quad m_{ij} \leftarrow a_{ij} + b_{ij} \rightarrow n \cdot n \rightarrow T(n) \cdot n^2$$

$$\quad \quad \text{endfor}$$

$$\quad \text{endfor}$$

$$\begin{array}{lll} * \text{ Big } O & * \text{ Big } \Omega & * \text{ Big } \Theta \\ n^3 \leq C n^3 & n^3 \geq C n^3 & \text{maka Big } \Theta \text{ bernilai sama} \\ 1 \leq C & 1 \geq C & \text{Big } \Theta = \Theta(n^3) \\ C \geq 1 & C \leq 1 & \end{array}$$

$$5) \text{ Algoritma menyilin link}$$

$$\text{for } i \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad a_i \leftarrow b_i \Rightarrow n = T(n)$$

$$\text{endfor}$$

$$\begin{array}{lll} * \text{ Big } O & * \text{ Big } \Omega & * \text{ Big } \Theta = \text{Big } \Omega, \text{ maka} \\ n \leq C n & n \geq C n & \text{Big } \Theta \text{ bernilai sama} \\ 1 \leq C & 1 \geq C & \text{Big } \Theta = \Theta(n) \\ C \geq 1 & C \leq 1 & \end{array}$$

$$6) a. \text{ jumlah operasi perbandingan}$$

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

$$b. \text{ Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan}$$

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

$$c. \text{ Hitung Kompleksitas}$$

$$\bullet \text{ Best case (semua telah terurut)}$$

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

$$\bullet \text{ Worst case (semua data harus ditukar)}$$

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

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

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

$$\text{Big } O$$

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

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

$$n_0 = 1 \rightarrow 2 - 2 \leq C$$

$$C \geq 0$$

$$\text{Big } \Omega$$

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

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

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

$$C \leq 0$$

$$7) a. \text{ Algoritma A} \rightarrow O(\log N)$$

$$b. \text{ Algoritma B} \rightarrow O(N \log N)$$

$$c. \text{ Algoritma C} \rightarrow O(N^2)$$

$$\text{Jika } N = 8 \text{ mana algoritma yang paling efektif?}$$

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

$$b. O(\log 8) = O(2 \log 2) \left\{ \begin{array}{l} \text{Kedua sama kecil} \\ O() \text{ maka efektif} \end{array} \right.$$

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

$$8) \text{ Operasi memasukkan nilai}$$

$$\bullet b_n \leftarrow a_n \quad 1 \text{ kali}$$

$$\bullet b_k \leftarrow a_k + b_k + 1 + x \quad n \text{ kali}$$

$$T(n) : n+1$$

$$O(n) = \text{urutan } p^2$$

$$\text{Algoritma P}$$

$$\text{Perpindahan } n \text{ kali}$$

$$\text{Perkalian } n \text{ kali}$$

$$T(n) = 2n$$

$$P^2 \text{ lebih baik dari P}$$