

LAPORAN PRAKTIKUM 3

ANALISIS ALGORITMA



DISUSUN OLEH:

NAMA : Putri Nabila
NPM : 140810180007

Program Studi S-1 Teknik Informatika
Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran
2020

Jawaban Tugas Latihan Analisa

1. Untuk $T(n) = 2 + 4 + 8 + 16 + \dots + n^2$, 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 $n \geq n_0$

Jawab :

$$\begin{aligned} T(n) &= 2 + 4 + 8 + 16 + \dots + 2^n \\ &= 2(2^n - 1) / (2 - 1) \\ &= 2(2^n - 1) \\ &= 2^{n+1} - 2 \end{aligned}$$

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

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

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

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

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

$$2 - 1 \leq C$$

$$C \geq 1$$

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

Jawab :

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

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

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

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

$$p + \frac{q}{n} + \frac{r}{n^2} \leq C \cdot n^2, n_0 = 1$$

$$p + q + r \leq C$$

$$C \geq p + q + r$$

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

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

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

$$pn + q + \frac{r}{n} \geq c$$

$$p + q + r \geq c$$

$$c \leq p + q + r$$

\Rightarrow Karena $\text{Big O} = \text{Big } \Omega = n^2$
maka, $\text{Big } \Theta = n^2$

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 ← 1 to n do
      wij ← wij or wik or
        wkj → n.n.n
    end for
  end for
end for

```

Waktu: $T(n) = n^3$

Big O

$$n^3 \leq c \cdot n^3$$

$$1 \leq c$$

$$c \geq 1$$

Big Ω



$$n^3 \geq c \cdot n^3$$

$$c \leq 1$$

Big Θ

$$\text{Big O} = \text{Big } \Omega$$

$$\text{Maka, Big } \Theta = \Theta(n^3)$$

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?

Algoritma Penjumlahan matriks $n \times n$

```

For i ← 1 to n do
  For j ← 1 to n do

```

```

    mij ← aij + bij → n.n → T(n) = n2
  end for
end for

```

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

end for

end for

Big O

$$n^2 \leq c \cdot n^2$$

$$1 \leq c$$

$$c \geq 1$$

Big Ω

$$n^2 \geq c \cdot n^2$$

$$1 \geq c$$

$$c \leq 1$$

Big Θ

$$\text{Big O} = \text{Big } \Omega$$

$$\text{Maka Big } \Theta = \Theta(n^2)$$

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?

Algoritma mengurutkan link

For $i \leftarrow 1$ to n do

$a_i \leq b_i \Rightarrow n = T(n)$

end for

Big O

$n \leq cn$

$1 \leq c$

$c \geq 1$

Big Ω

$n \geq cn$

$1 \geq c$

$c \leq 1$

Big Θ

$Big O = Big \Omega$

$Big \Theta = \Theta(n)$

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  $\leftarrow 1$  to  $n - 1$  do
    for k  $\leftarrow n$  downto pass + 1 do
      if  $a_k < a_{k-1}$  then
        { pertukarkan  $a_k$  dengan  $a_{k-1}$  }
        temp  $\leftarrow a_k$ 
         $a_k \leftarrow a_{k-1}$ 
         $a_{k-1} \leftarrow temp$ 
      endif
    endfor
  endfor

```

- Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
- Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
- Hitung kompleksitas waktu asimtotik (Big-O, Big- Ω , dan Big- Θ) dari algoritma Bubble Sort tersebut!

a) jumlah operasi perbandingan
 $1+2+3+4+\dots+(n-1)$
 $= \frac{n(n-1)}{2}$ kali

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

c) • Best case (semua sudah terurut dengan baik)
 $\frac{(n-1)}{2} n$ kali. $T_{\min}(n) = \frac{n(n-1)}{2} = \frac{n^2-n}{2}$
 • Worst case (data yang harus diiterasi semua)
 Perbandingan $\rightarrow \frac{n(n-1)}{2}$
 masukkan nilai $\rightarrow \frac{3n(n-1)}{2}$
 $T_{\max}(n) = \frac{4n(n-1)}{2} = 2n^2 - 2n$

Big O
 $2n^2 - 2n \leq cn^2$
 $2 - \frac{2}{n} \leq c \rightarrow n \geq 1$
 $2 - 2 \leq c$
 $c \geq 0$

Big Ω
 $\frac{n^2-n}{2} \geq cn^2$
 $\frac{1}{2} - \frac{1}{2n} \geq c, n \geq 1$
 $\frac{1}{2} - \frac{1}{2} \geq c$
 $c \leq 0$

7. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:

- Algoritma A mempunyai kompleksitas waktu $O(\log N)$
- Algoritma B mempunyai kompleksitas waktu $O(N \log N)$
- 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?

a) algoritma A $\rightarrow O(\log N)$
 b) Algoritma B $\rightarrow O(N \log N)$
 c) Algoritma C $\rightarrow O(N^2)$
 jika $N=8$ maka algoritma paling efektif adalah.
 kita uji:
 a) $O(\log 8) = O(3 \log_2)$
 b) $O(8 \log 8) = O(24 \log_2)$
 c. $O(8^2) = O(64)$
 maka yang paling efektif adalah algoritma A karena semakin kecil $O()$, semakin efektif

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
 b₁, b₂, ..., b_n : real

Algoritma

b_n ← a_n
 for k ← n - 1 downto 0 do
 b_k ← a_k + b_{k+1} * x
 endfor
return b₀

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?

② Operasi memasukkan nilai p.
 → b_n ← a_n (1 kali)
 → b_k ← a_k + b_{k+1} * x (n kali)
 T(n) = n + 1
 O(n) = untuk p₁
 algoritma p → Penjumlahan n kali
 Perkalian n kali
 maka, p² lebih baik dari p.
