TUGAS MODUL PRAKTIKUM 3

Disusun oleh :

Anne Audistya Fernanda

140810180059

Kelas A

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2020

Latihan Analisa

Minggu ini kegiatan praktikum difokuskan pada latihan menganalisa, sebagian besar tidak perlu menggunakan komputer dan mengkoding program, gunakan pensil dan kertas untuk menjawab persoalan berikut!

1. Untuk $T(n) = 2 + 4 + 8 + 16 + \cdots + n^2$, tentukan nilai C, $f(n), n_0$, dan notasi Big-O sedemikian sehingga $T(n) = O\big(f(n)\big)$ jika $T(n) \le C\ untuk\ semua\ n \ge n_0$
   Jawab :

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

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

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

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

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

$2 \cdot 2^n - 2 \le C \cdot 2^n$ .... dibagi 2^n

$2 - \dfrac{2}{2^n} \le C$ ... $n_0 = 1$

$2 - \dfrac{2}{2} \le C$

$C \ge 1$

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 :

$\boxed{2}$ $T(n) = pn^2 + qn + r$

- $O(n^2) \Rightarrow$ Big O

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

$pn^2 + qn + r \le C \cdot n^2$ .... dibagi n^2

$p + \dfrac{q}{n} + \dfrac{r}{n^2} \le C$ .... $n_0 = 1$

$p + q + r \le C$

$C \ge p + q + r$

- $\Omega(n^2) \Rightarrow$ Big Ω

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

$pn^2 + qn + r \ge C \cdot n^2$ .... dibagi n^2

$p + \dfrac{q}{n} + \dfrac{r}{n^2} \ge C$ .... $n_0 = 1$

$p + q + r \ge C$

$C \le p + q + r$

- Karena Big O = Big Ω = n^2,
  maka Big Θ = 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 ← w_ij or w_ik and w_kj
        endfor
    endfor
endfor
```

Jawab :

```
[3] for k ← 1 to n do
      for i ← 1 to n do
        for j ← to n do
          Wij ← Wij or Wik or Wkj ⇒ n . n . n
        endfor                      T(n) = n³
      endfor
    endfor
```

▸ Big O

$n^3 \le C \cdot n^3$ .... dibagi n^3

$1 \le C$

$C \ge 1$

▸ Big Ω

$n^3 \ge C \cdot n^3$ .... dibagi n^3

$1 \ge C$

$C \le 1$

▸ Big Θ

Karena Big O = Big Ω = n^3,

maka Big Θ = $\Theta(n^3)$

4. Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran n x n. Berapa kompleksitas waktunya T(n)? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big-Ω, dan Big-Θ?

Jawab :

[4] Algoritma penjumlahan matriks n x m

```
for i ← 1 to n do
  for j ← 1 to n do
    mij ← aij + bij ⇒ n . n
  endfor                 T(n) = n²
end for
```

▸ Big O

$n^2 \le C \cdot n^2$ .... dibagi n^2

$C \ge 1$

▸ Big Ω

$n^2 \ge C \cdot n^2$ .... dibagi n^2

$C \le 1$

▸ Big Θ

Karena Big O = Big Ω = n^2,

maka big Θ = $\Theta(n^2)$

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?
   Jawab :

5 Algoritma menyalin larik

<u>for</u> i ← 1 to n do
  $a_i \leftarrow b_i \Rightarrow n = T(n)$
<u>endfor</u>

- Big O
  $n \le C \cdot n$ ... dibagi n
  $C \ge 1$
- Big Ω
  $n \ge C\ n$ ... dibagi n
  $C \le 1$
- Big θ
  Karena Big O = Big Ω = n
  maka Big θ = θ(n)

6. Diberikan algoritma Bubble Sort sebagai berikut:

```
procedure BubbleSort(input/output a1, a2, ..., an : integer)
{ Mengurut tabel integer TabInt[1..n] dengan metode pengurutan bubble-
sort
  Masukan: a1, a2, ..., an
  Keluaran: a1, a2, ..., an (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 ak < ak-1 then
            { pertukarkan ak dengan ak-1 }
            temp ← ak
            ak ← ak-1
            ak-1←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!
      Jawab :

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 terurut)

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

$= \underline{n^2 -}$

- Worst Case (semua tertukar)

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

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

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

Big O

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

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

$2 - 2 \leq C$

$C \geq 0$

Big Ω

$\frac{n^2 - n}{2} \geq C \cdot n^2 \ldots$ dibagi n^2

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

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

$C \leq 0$

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)

   Untuk problem X dengan ukuran N=8, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

   Jawab :

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) $O(\log 8) = O({}^3\log 2)$

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

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

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:

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

```
function p2(input x : real) → real
{ Mengembalikan nilai p(x) dengan metode Horner}

Deklarasi
  k : integer
  b₁, b₂, ..., bₙ : real

Algoritma
  bₙ ← aₙ
  for k ← n - 1 downto 0 do
     bₖ ← aₖ + bₖ₊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?

Jawab :

$\boxed{8}$ Operasi memasukkan nilai

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

$T(n) = n + 1$

$O(n)$ = untuk p^2

Algoritma p

Penjumlahan n kali

Perkalian n kali

$T(n) = 2n$

p^2 lebih baik dari p