LAPORAN PRAKTIKUM 3

ANALISIS ALGORITMA

Disusun oleh :

Hafidh Akhdan Najib

140810180061

PROGRAM STUDI S-1 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 $T(n) = 2 + 4 + 6 + 8 + 16 + \cdots + n^2$, tentukan nilai C, f(n), n_o, dan notasi Big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C\ untuk\ semua\ n\ \geq n_0$

Jawab :

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

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

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

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

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

$$2.2^n - 2 \leq C.2^n$$

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

nilai $n_0 : 1$

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

$$C \geq 1$$

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

Jawab :

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

$O(n^2) \rightarrow$ Big O

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

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

$p + \frac{q}{n} + \frac{r}{n^2} \leq c$

$n_0 = 1$ misal

$p + q + r \leq c$

$c \geq p + q + r$

$\Omega(n^2) \rightarrow$ Big

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

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

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

misal $n_0 = 1$

$p + q + r \geq c$

$c \leq p + q + r$

Big $\theta(n^2)$

Karena keduanya berderajat sama maka $\theta(n^2)$ terbukti benar

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 and wkj    → n.n.n
            end for                            T(n) = n^3
        endfor
    endfor
```

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

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

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.) for i ← 1 to n do
for j ← 1 to n do
$m_{i,j} \leftarrow a_{i,j} + b_{ij} \rightarrow n \cdot n$
$T(n) = n^2$
endfor
endfor

• Big O
$n^2 \le c \cdot n^2$
$1 \le c$
$c \ge 1$

• Big Ω
$n^2 \ge c \cdot n^2$
$1 \ge c$
$c \le 1$

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

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.) for i ← 1 to n do
$a_i \leftarrow b_i \rightarrow n \rightarrow T(n) = n$
endfor

• Big O
$n \le c \cdot n$
$1 \le c$
$c \ge 1$

• Big Ω
$n \ge cn$
$1 \ge c$
$c \le 1$

• Big θ
karena $O(n)$ dan $\Omega(n)$ berderajat sama, maka $\theta(n)$

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

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

(b.) Maksimum pertukaran elemen tabel

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

(c) Kompleksitas

- Best case

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

- Worst case

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

Assignment $\rightarrow \underline{3n(n-1)}$

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

• Big O

$2n^2-2n \leq c(n^2)$

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

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

$0 \leq c$

• Big Ω

$\frac{n^2-n}{2} \geq c\,n^2$

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

$n_0 - 1 \rightarrow 1/2 - 1/2 \geq c$

$c \leq 0$

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 $O(N^2)$
    Untuk problem X dengan ukuran N=8, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

Jawab :

7.) Misal $N = 8$

maka algoritma

a. $A \rightarrow O(\log 8) = O({}^{3}\log 2)$

b. $B \rightarrow O({}^{8}\log 8) = O({}^{2^n}\log 2)$

c. $C \rightarrow O(8^2) = O(64)$

Jadi, Algoritma A paling efektif karena semakin kecil nilai dalam O < 1 semakin baik operasi yang digunakan

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 :

8) Operasi Assignment

- $b_n \leftarrow a_n$ = 1 kali
- $b_n \leftarrow a_n + b_{n-1} \cdot x$ = n kali

T (n) = n + 1

O (n) untuk p2

Algoritma P

Penjumlahan : n kali

Perkalian : n kali

T (n) = 2n

Algoritma p2 lebih baik dari pd P