LAPORAN PRAKTIKUM 3
ANALISIS ALGORITMA

DISUSUN OLEH:

NAMA : RAISSA AMINI

NPM : 140810180073

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

2018

Latihan Analisa

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 $T(n) \leq C \; untuk \; semua \; n \geq n_0$

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

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

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

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

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

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

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

   $2 - 1 \leq C$

   $C \geq 1$

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

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

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

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

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

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

   $p + q + r \leq C$

   $C \geq p + q + r$

   -) $\Omega(n^2) \rightarrow$ Big Ω

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

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

   $pn + q + \frac{r}{n} \geq C$ jika $n_0 = 1$

   $p + q + r \geq C$

   $C \leq p + q + r$

   -) karena Big O = Big $\Omega = n^2$

   maka 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 ← to n do
            w_ij ← w_ij or w_ik and w_kj
        endfor
    endfor
endfor
```

3.) For k ← i to n do
For i ← 1 to n do
For j ← 1 to n do
wij ← wij or wik or wkj ⇒ n.n.n
T(n) = n^3
end For
end For
end For

•) Big O
$n^3 \leq C.n^3$
$1 \leq C$
$C \geq 1$

•) Big Ω
$n^3 \geq C.n^3$
$C \leq 1$

•) Big θ
Big O = Big Ω
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?

A) Algoritma penjumlahan matriks n x m
For i ← i to n do
For j ← 1 to n do
mij ← aij + bij ⇒ n.n ⇒ T(n) = n^2
end For
end For

•) Big O
$n^2 \leq C\,n^2$
$1 \leq C$
$C \geq 1$

•) Big O = Big Ω
maka Big θ = $\theta(n^2)$

•) Big Ω
$n^2 \geq C.n^2$
$1 \geq C$
$C \leq 1$

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?

5.) Algoritma menyalin larik

```
For i ← i to n do
    ai ← bi => n = T(n)
endFor
```

a.) Big O

$n \leq Cn$

$1 \leq C$

$C \geq 1$

b.) Big Ω

$n \geq Cn$

$1 \geq C$

$C \leq 1$

c.) Jika Big O = Big Ω

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!

6.) a) Jumlah operasi perbandingan
$1+2+3+4+\cdots+(n-1)$
$= \frac{n(n-1)}{2}$ kali

b.) berapa kali maksimum pertukaran elemen - elemen tabel dilakukan
$\frac{n(n-1)}{2}$ kali

c.) hitung kompleksitas
- Best Case (semua telah terurut)
$\frac{(n-1)n}{2}$ kali . $T\min(n) = \frac{n(n-1)}{2} = \frac{n^2-n}{2}$
- worst case (semua data harus ditukar)
Perbandingan → $\frac{n(n-1)}{2}$
memasukkan nilai → $\frac{3n(n-1)}{2}$
$T\max(n) = \frac{4n(n-1)}{2} = 2n^2-2n$

- Big O
$2n^2-2n \leq C n^2$
$2-\frac{2}{n} \leq C \rightarrow n_0 = 1$
$2-2 \leq C$
$C \geq 0$

- Big Ω
$\frac{n^2-n}{2} \geq C n^2$
$\frac{1}{2}-\frac{1}{2}n \geq C \rightarrow n_0 = 1$
$\frac{1}{2}-\frac{1}{2} \geq C$
$C \leq 0$

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?

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

8.) Operasi memasukkan nilai

- bn ← an 1 kali
- bn ← ak+bk+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