LAPORAN PRAKTIKUM

ANALISIS ALGORITMA

Alfari Sidnan Ghilmana

140810180011

Kelas A

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

Kerjakan di lab, worksheet pada Modul Praktikum 3 di Google Classroom

Worksheet 3

1. Untuk $T(n) = 2 + 4 + 6 + 8 + 16 + \cdots + n^2$, 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 ← w_ij or w_ik and w_kj
        endfor
    endfor
endfor
```

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?

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 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!

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:

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

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

Deklarasi
  k : integer
  b0, b1, ..., 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?

Nama : Alfari Sidnan G
NPM : 1408101800111
Kelas : A

Worksheet 3

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

Deret Geometri $= \frac{a(r^n-1)}{r-1} = \frac{2(2^n-1)}{2-1}$

$= 2(2^n-1)$

$= 2^{n+1}-2$

Notasi big $O(n) \rightarrow O(2^n)$

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

$2^{n+1}-2 \leq c.2^n$ misal $n_0 = 1$

$\frac{2^{n+1}}{2^n} - \frac{2}{2^n} \leq c$ $2 - \frac{2}{2} \leq c$

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

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

→ Big $O(n^2)$

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

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

$\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \leq \frac{c.n^2}{n^2}$

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

misalkan $n_0 = 1$

misalkan $p = q = r = 1$

$1 + \frac{1}{1} + \frac{1}{1} \leq c$

$c \geq 3$ .. Terbukti / benar

→ Big $\Omega(\Omega(n^2))$

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

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

$\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \geq c$

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

misalkan $n_0 = 1$

$p + q + r \geq c$

misalkan $p = 1, q = 1$, $r = 1$

$1+1+1 \geq c$

$c \leq 3$ .. terbukti / benar

→ Big Θ

karena $O(n^2)$ dan $\Omega(n^2)$ terbukti dan berderajat sama maka $\Theta(n^2)$ terbukti benar

3.) Kompleksitas waktu

Operasi Assigment

$w_{ij} \leftarrow w_{ik}$ or w_{ik} and w_{kj} berulang sebanyak

n kali di loop "for j ← i to n do" serta

n kali di loop "for i ← 1 to n do" dan

n kali di loop "for k ← 1 to n do" maka

$T(n) = n.n.n = n^3$

→ Big $O()$ $O(n^3)$

$n^3 \leq c.n^3$

$\frac{n^3}{n^3} \leq c$

$1 \leq c$

$c \geq 1$

→ Big Ω

$n^3 \geq c\, n^3$

$1 \geq c$

$c \leq 1$

→ Big Θ : $\Theta(n^3)$

karena $O(n^3)$ dan $\Omega(n^3)$ berderajat sama maka $\Theta(n^3)$

4.) Algoritma menjumlahkan dua matriks

```
for i ← 1 to n do
  for j ← 1 for n do
    mij ← aij + bij
  end for
endfor
```

$T(n) = n^2$

→ $O(n^2)$: $n^2 \leq cn^2$, $c \geq 1$

→ $\Omega(n^2)$: $n^2 \geq cn^2$, $c \leq 1$

→ $\Theta(n^2)$: $O(n^2)$ dan $\Omega(n^2)$ berderajat sama maka $\Theta(n^2)$

5.) Algoritma menyalons larik

```
for i ← 1 to n do
  ai ← bi
endfor
```

$T(n) = n$

→ $O(n)$: $n \leq c.n$, $c \geq 1$

→ $\Omega(n)$: $n \geq c.n$, $c \leq 1$

→ $\Theta(n)$: $O(n)$ dan $\Omega(n)$ derajat sama maka $\Theta(n)$

a. Jumlah operasi perbandingan

$T(n) = (n-1) + (n-2) + (n-3) + \dots + 1$

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

b.) maksimum perluasan terjadi ketika $\frac{n(n-1)}{2}$

c.) kompleksitas waktu

-> Best case

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

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

-> Worst case

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

Assigment $\rightarrow \frac{3n(n-1)}{2}$ kali

$T_{max}(n) = \frac{n(n-1)}{2} + \frac{3n(n-1)}{2} = \frac{4n(n-1)}{2} = 2n^2 - 2$

-> $O(n^2)$

$2n^2 - 2 \leq c.n^2$

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

$c \geq 0$

$\Omega(n^2)$
$2n^2 - 2 \geq c.n^2$
$2 - \frac{2}{n^2} \geq c, n_0 = 1$
$c \leq 0$

-> $\Theta(n^2)$

$O(n^2)$ dan $\Omega(n^2)$ berderajat sama

7.) a.) Algoritma A -> $O(\log N)$

b.) Algoritma B -> $O(\log N)$

c.) Algoritma C -> $O(n^2)$

$N = 8$ maka

algoritma A -> $O(\log 8) = O(3 \log 2)$

algoritma B -> $O(8 \log 8) = O(24 \cdot \log 2)$

algoritma C -> $O(8^2) = O(64)$

Algoritma A lebih cepat dan no 1 yg lainya

8.) Operasi Assigment

-> $bn \leftarrow an$ - 1 kali

-> $hu \leftarrow ak + bu \times$: n kali

$T(n) = n+1$

$O(n)$ untuk P'

Algoritma P

Pertambahan : n kali

Perkalian : n kali

$T(n) = 2n$

maka algoritma P' lebih baik dari pada P