LAPORAN PRAKTIKUM
ANALISIS ALGORITMA

DISUSUN OLEH

SALMA ALIFIA SHAFIRA 140810180058

UNIVERSITAS PADJADJARAN

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

TEKNIK INFORMATIKA

2020

Nama : Salma Alifa Shafira
NPM : 140810180058
Kelas : B

LATIHAN ANALISA

① Untuk $T(n) = 2+4+6+8+16+\dots+n^2$
Tentukan : c, f(n), n_0, notasi Big-O
Bentuk deret geometri :

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

Notasi Big O ⇒ $O(2^n)$
$T(n) \leq c \cdot 2^n$ → $c \geq 1$
$2^{n+1}-2 \leq c \cdot 2^n$
$\frac{2^{n+1}}{2^n} - \frac{2}{2^n} \leq c$
$2 - \frac{2}{2^n} \leq c$, misal $n_0 = 1$

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

- Big O ($O(n^2)$)
  $T(n) \leq c \cdot f(n)$
  $pn^2+qn+r \leq c \cdot n^2$
  $\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \leq c$, misal $n_0 = 1$
  $p + q + r \leq c$, misal $p, q, r = 1$
  $c \geq 3$
- Big-Ω ($\Omega(n^2)$)
  $T(n) \geq c(g(n))$
  $pn^2+qn+r \geq c \cdot n^2$
  $\frac{pn^2}{n^2} + \frac{qn}{n^2} + \frac{r}{n^2} \geq c$, misal $n_0 = 1$
  $c \leq p+q+r$, misal $p, q, r = 1$
  $c \leq 3$
- Big Θ ($\Theta(n^2)$)
  Karena $O(n^2)$ dan $\Omega(n^2)$ benar dan berderajat sama maka $\Theta(n^2)$ terbukti benar

③ Kompleksitas Waktu asimptotik Big O, Ω, Θ.
Wij ← Wij or Wik and Wkj → n^3
$T(n) = n^3$

- Big O → $O(n^3)$
  $n^3 \leq c \cdot n^3$
  $c \geq 1$
- Big Ω → $\Omega(n^3)$
  $n^3 \geq c \cdot n^3$
  $c \leq 1$
- Big Θ → $\Theta(n^3)$
  ⇒ $O(n^3)$ & $\Omega(n^3)$ berderajat sama. maka $\Theta(n^3)$ benar

④ Algoritma menjumlahkan dua matriks

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

$T(n) = n \cdot n = n^2$

- $O(n^2)$
  $n^2 \leq c \cdot n^2$
  $c \geq 1$
- $\Omega(n^2)$
  $n^2 \geq c \cdot n^2$
  $c \leq 1$
- $\Theta(n^2)$
  ⇒ $O(n^2)$ & $\Omega(n^2)$ berderajat sama
  Maka $\Theta(n^2)$ benar

⑤ Algoritma menjumlahkan Waktu

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

$T(n) = n$

- OΩ
  $n \leq c \cdot n$
  $c \geq 1$
- $\Omega(n)$
  $n \geq c \cdot n$
  $c \leq 1$
- $\Theta(n)$
  ⇒ $O(n)$ dan $\Omega(n)$ berderajat sama
  Maka $\Theta(n)$ benar

⑥ a) Jumlah Operasi perbandingan
$0+1+2+3+4+\dots+(n-1)\times = \frac{n(n-1)}{2}$ kali

b) Berapa kali pertukaran elemen² tabel dilakukan (maksimum) ! → $\frac{n(n-1)}{2}$ kali

c) Hitung kompleksitas waktu asimptotik
BEST CASE → data sudah terurut
perbandingan $\frac{n(n-1)}{2}$ kali
$T_{min}(n) = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$
WORST CASE → Data harus ditukar/urut.
Perbandingan $\frac{n(n-1)}{2}$ kali
Assignment → $\frac{3n(n-1)}{2}$ kali
$T_{max}(n) = \frac{4n(n-1)}{2} = 2n^2-2n$

- $O(n^2)$
  $2n^2-2n \leq c \cdot n^2$
  $2 - \frac{2}{n} \leq c$, misal $n_0 = 1$
  $c \geq 2-2$
  $c \geq 0$
- $\Omega(n^2)$
  $\frac{n^2}{2} - \frac{n}{2} \geq c \cdot n^2$
  $\frac{1}{2} - \frac{1}{2n} \geq c$, misal $n_0 = 1$
  $\frac{1}{2} - \frac{1}{2} \geq c$
  $c \leq 0$
- $\Theta(n^2)$
  $O(n^2)$ & $\Omega(n^2)$ berderajat sama

⑦ a) algoritma A → $O(\log N)$

b) algoritma B → $O(N \log N)$

c) algoritma C → $O(N^2)$

N=8, maka

algoritma A → $O(\log 8) = O(3 \cdot \log 2)$

algoritma B → $O(8 \log 8) = O(24 \log 2)$

algoritma C → $O(8^2) = O(64)$

Dengan asumsi $\log 2 = 0{,}301$,

maka algoritma A lebih cepat dari pada B & C.

⑧ Operasi assignment

$b_n \leftarrow a_n$: 1 kali

$b_k \leftarrow a_k + b_{k+1} * x$: n kali

$T(n) = n+1$

$O(n)$ untuk P2

Algoritma P

Pertambahan : n kali

Perkalian : n kali

$T(n) = 2n$

Maka algoritma P2 lebih baik daripada P.