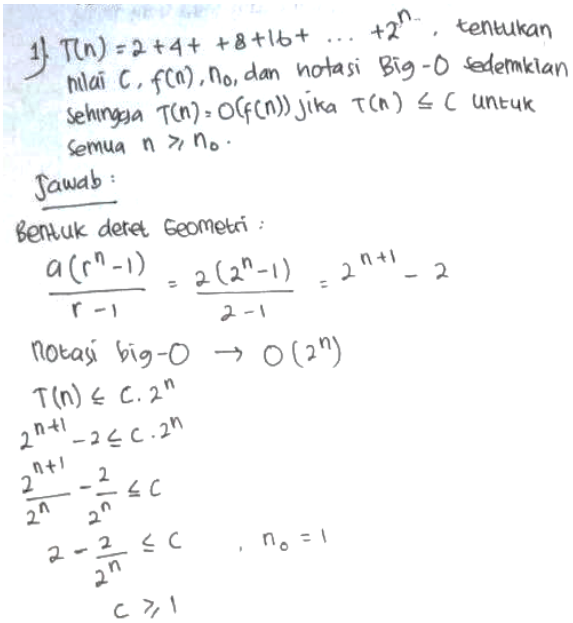
LAPORAN PRAKTIKUM 3
Analisis algoritma

Sarah Navianti Dwi Sutisna
140810180021
Kelas A

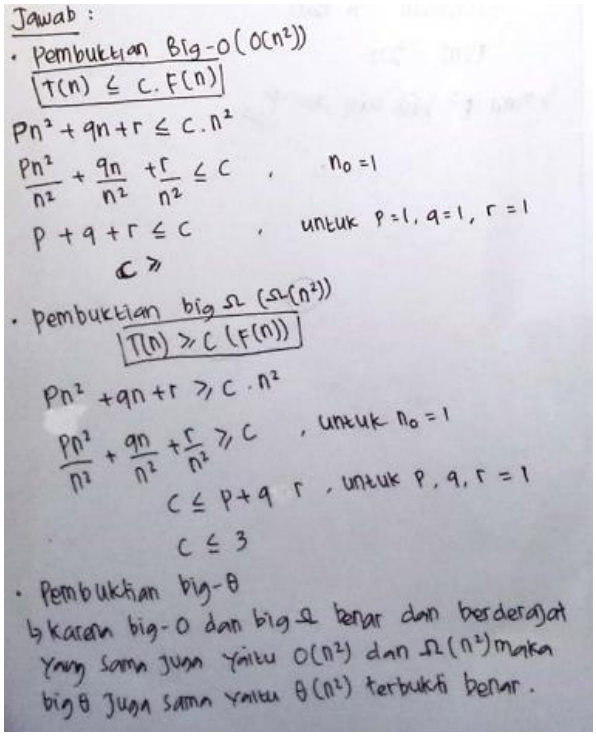
PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2020

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



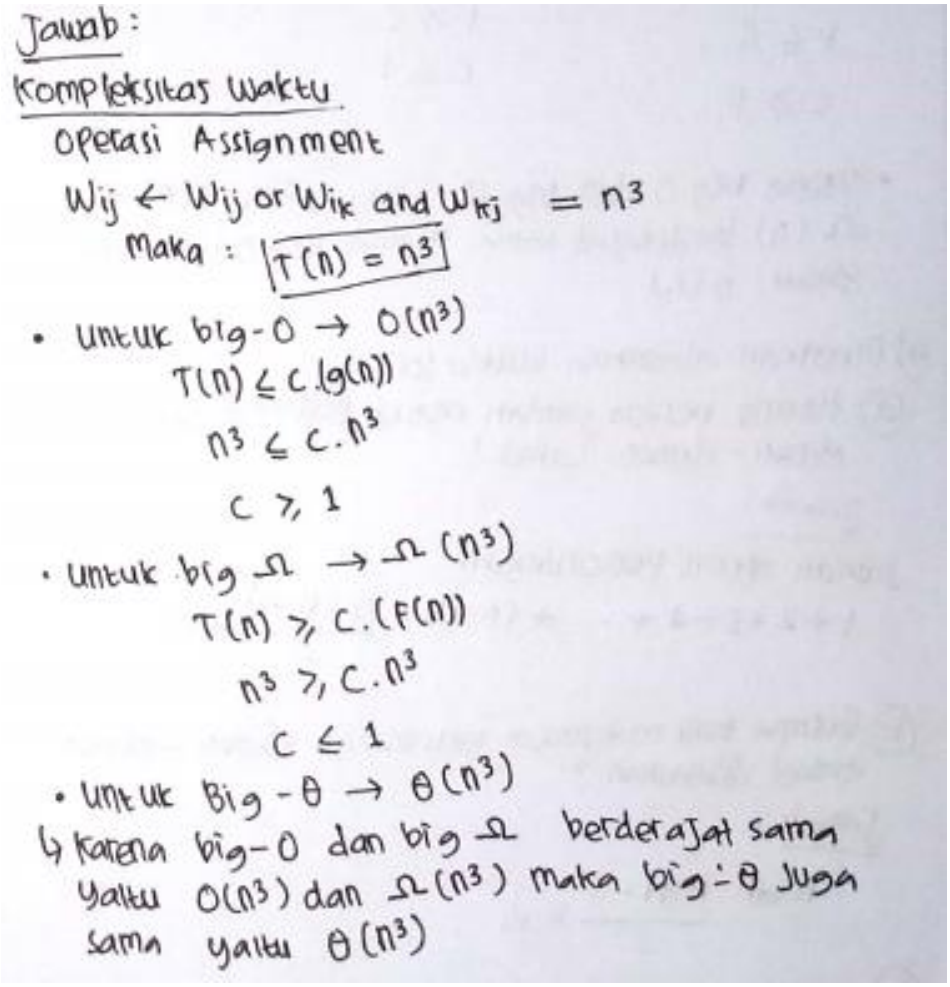
2. Buktikan bahwa untuk konstanta-konstanta positif p, q, dan r:
   $T(n) = pn^2 + qn + r$ adalah $O(n^2), \Omega(n^2), \Theta(n^2)$
   Jawab :



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 :



Jawab :

Kompleksitas Waktu

Operasi Assignment

$W_{ij} \leftarrow W_{ij}$ or W_{ik} and $W_{kj} = n^3$

Maka : $\boxed{T(n) = n^3}$

- Untuk big-O → $O(n^3)$

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

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

  $c \geq 1$

- Untuk big Ω → $\Omega(n^3)$

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

  $n^3 \geq c.n^3$

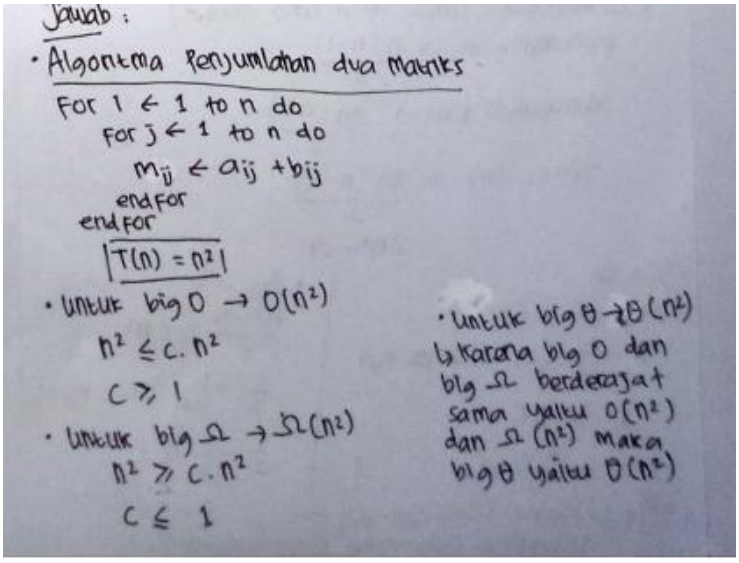
  $c \leq 1$

- Untuk Big-Θ → $\Theta(n^3)$

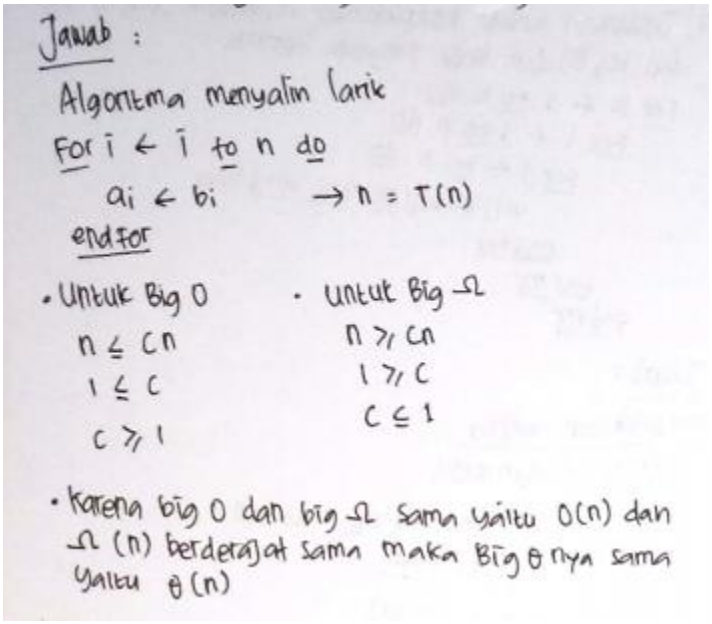
↳ karena big-O dan big Ω berderajat sama yaitu $O(n^3)$ dan $\Omega(n^3)$ maka big-Θ juga sama yaitu $\Theta(n^3)$

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 :



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?

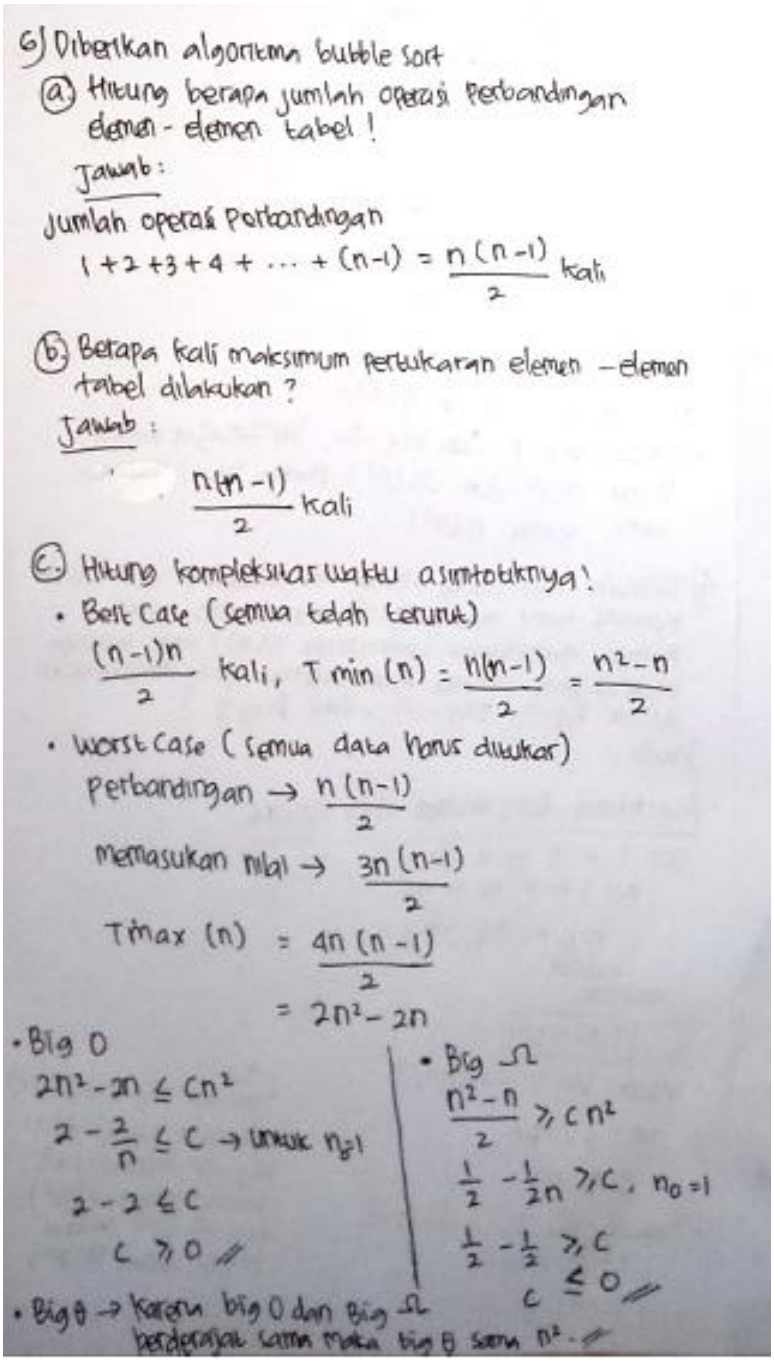


6. Diberikan algoritma Bubble Sort sebagai berikut:

```
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
```

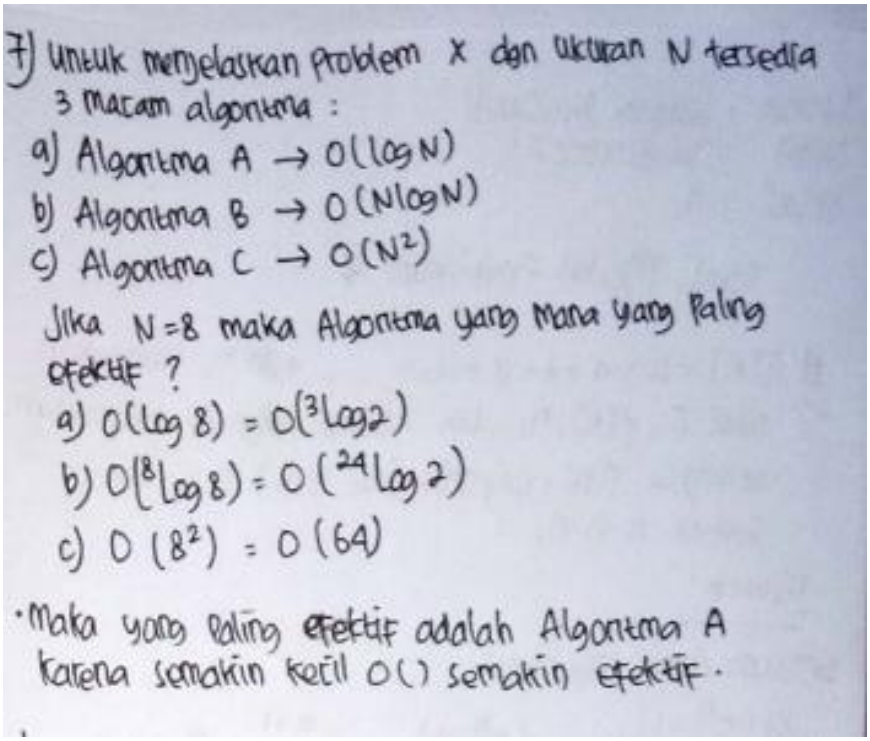
a. Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
b. Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
c. Hitung kompleksitas waktu asimptotik (Big-O, Big-Ω, dan Big-Θ) dari algoritma Bubble Sort tersebut!

Jawab :



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?



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 :