LAPORAN PRAKTIKUM

ANALISIS ALGORITMA

Disusun oleh:

Fadlan Mulya Priatna

140810180041

Kelas A

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

1. **Materi**

Definisi notasi BIG-O

, artinya berorde paling besar bila terdapat konstanta dan untuk

Jika dibuat semakin besar, waktu yg dibutuhkan tidak akan melebihi dikalikan . Maka adalah upper bound.

Dalam pembuktian Big-O, diperlukan nilai dan nilai agar terpenuhi kondisi

Notasi Polinomial BIG-O Berderajat

Digunakan untuk memperkirakan kompleksitas dengan mengabaikan suku berorde rendah. Contoh dinyatakan pada:

|  |
| --- |
| Teorema 1  Bila adalah polinom berderajat m maka |

Artinya:

Mengambil suku paling tinggi derajatnya yang diartikan laju pertumbuhan lebih cepat dibandingkan yang lainnya.

|  |
| --- |
| Teorema 2  Misalkan dan , maka   1. (i)   (ii)   1. , c adalah konstanta |

Contoh:

1. Misalkan , , dan , dengan m sebagai peubah, maka:
   1. Teorema 2(a)(i)
   2. Teorema 2(a)(ii)
   3. Teorema 2(b)
2. (a) Teorema 2(c)

(b) Teorema 2(d)

Aturan Kompleksitas Waktu Asimptotik

Cara 1

Jika kompleksitas waktunya sudah dihitung, maka kompleksitas waktu asimptotik dapat ditentukan dengan mengambil suku yang mendominasi fungsi & menghilangkan koefisiennya (Teorema 1)

Contoh:

Pada algoritma cariMax,

Cara 2

Big-O Notation:Pengisian nilai, perbandingan, operasi aritmatika, read, write, pengaksesan elemen larik, memilih field dari record, dan pemanggilan fungsi membutuhkan waktu . Cara penghitungan :

, dengan syarat nilai dan positif

Contoh

read (x) O(1)

x <- x+1 O(1) + O(1) = O(1)

write (x) O(1)

Kompleksitas waktu asimtotik algoritmanya O(1) + O(1) + O(1) = O(1)

Penjelasan :

O(1) + O(1) + O(1) = O(max(1,1)) + O(1)

= O(1) + O(1)

= O(max(1,1))

= O(1)

Notasi Big- Ω dan Big-Θ

Big-O hanya menyediakan upper bound.

Sedangkan untuk lower bound, dapat diperoleh dengan big-Ω dan big- Θ

Definisi

Notasi Big- Ω:

, artinya berorde paling kecil bila terdapat konstanta dan sehingga :

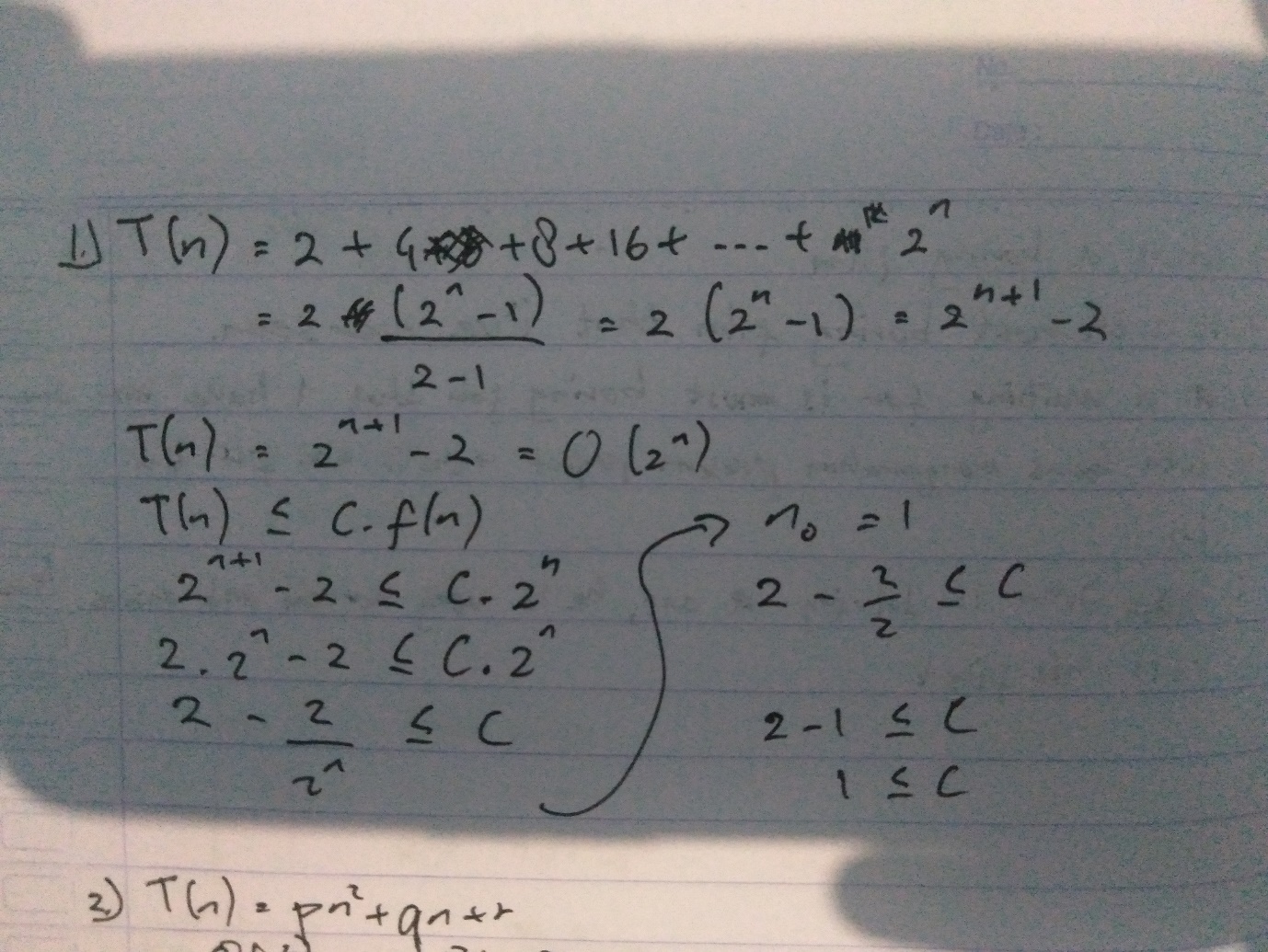
, dengan syarat nilai dan positif

Notasi Big- Ω:

, artinya berorde sama dengan Jika dan .

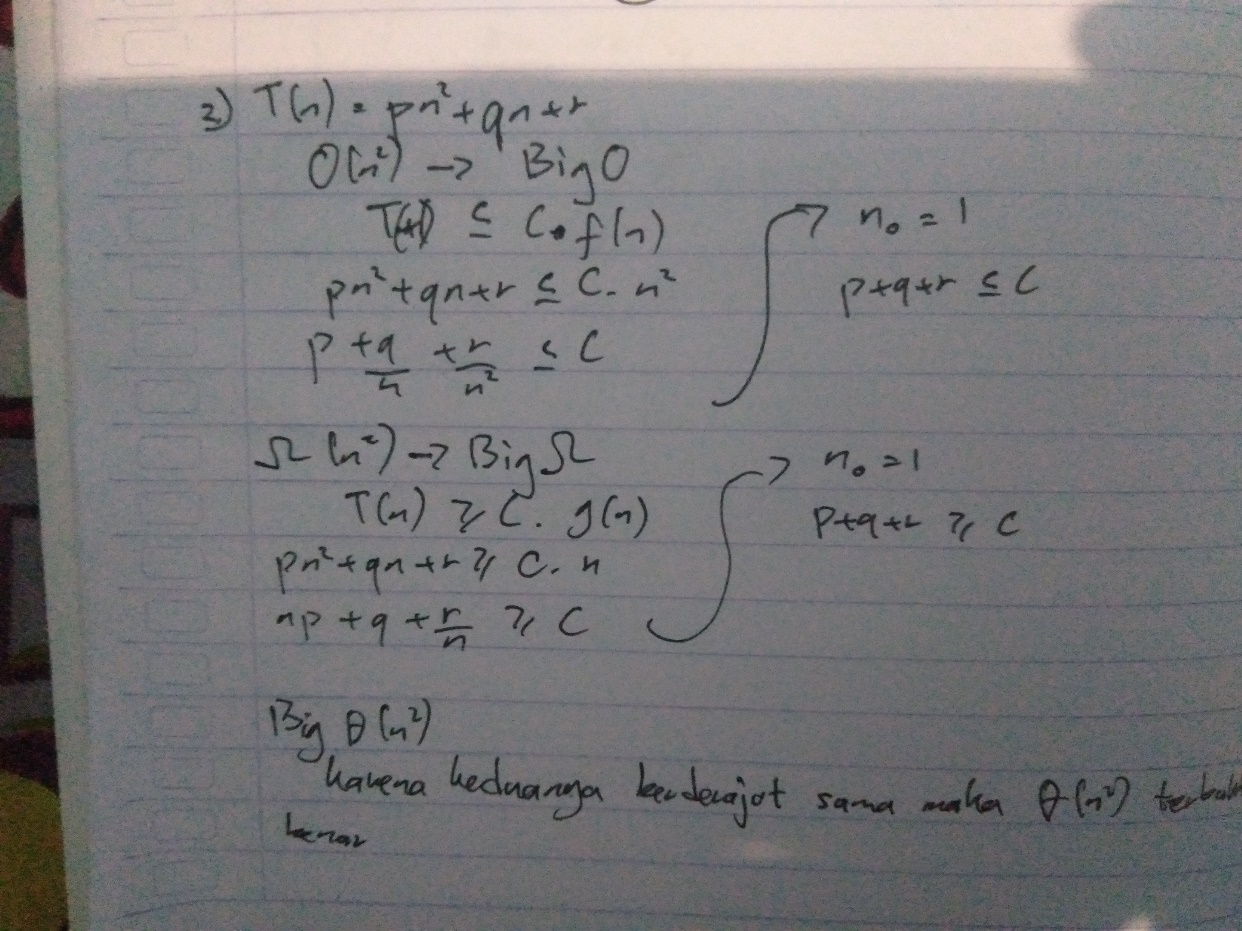
, dengan syarat nilai dan positif

1. **Bagian Analisis di Modul Praktikum**
2. Untuk , tentukan nilai dan notasi Big-O sedemikian sehingga jika untuk semua



1. Buktikan bahwa untuk konstanta-konstanta positif p, q, dan r:

adalah



1. 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

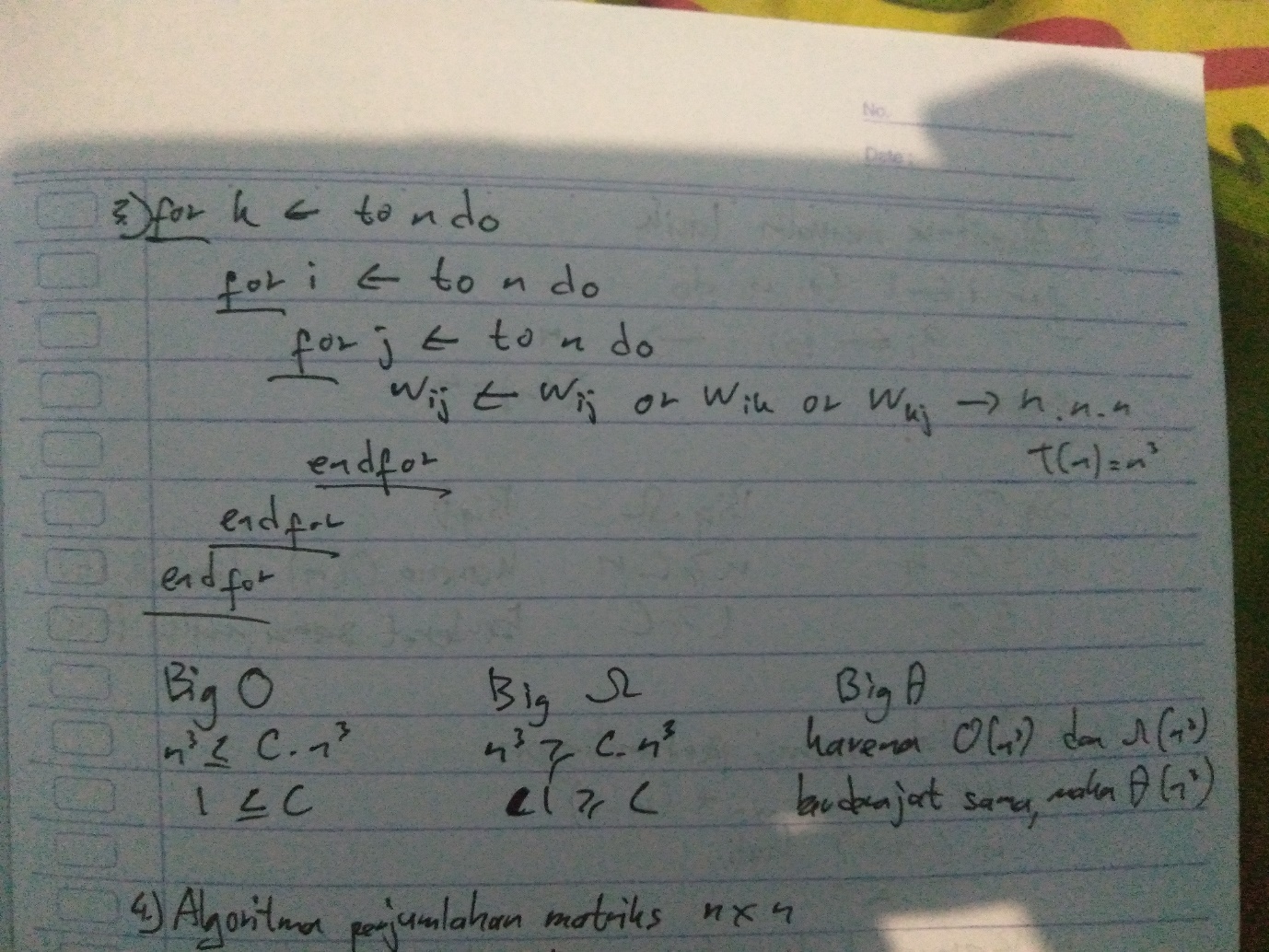
for j ← to n do

wij ← wij or wik and wkj

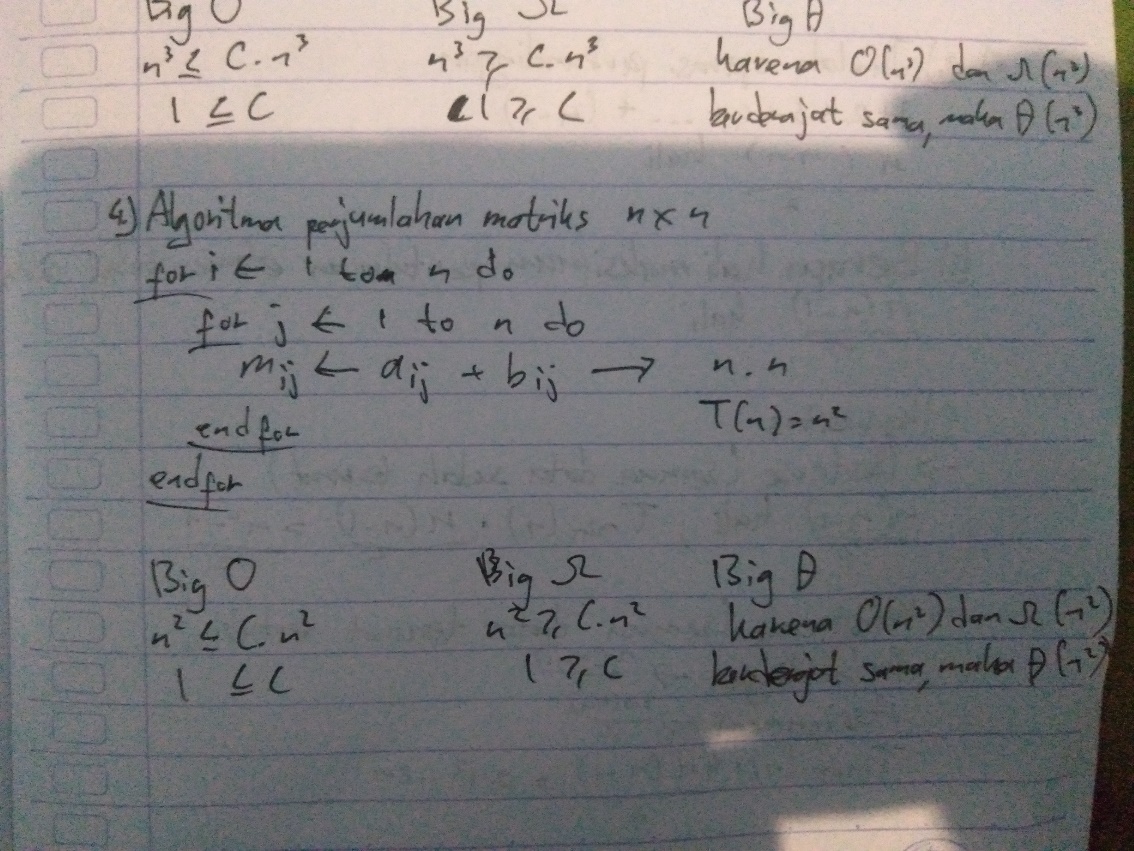
endfor

endfor

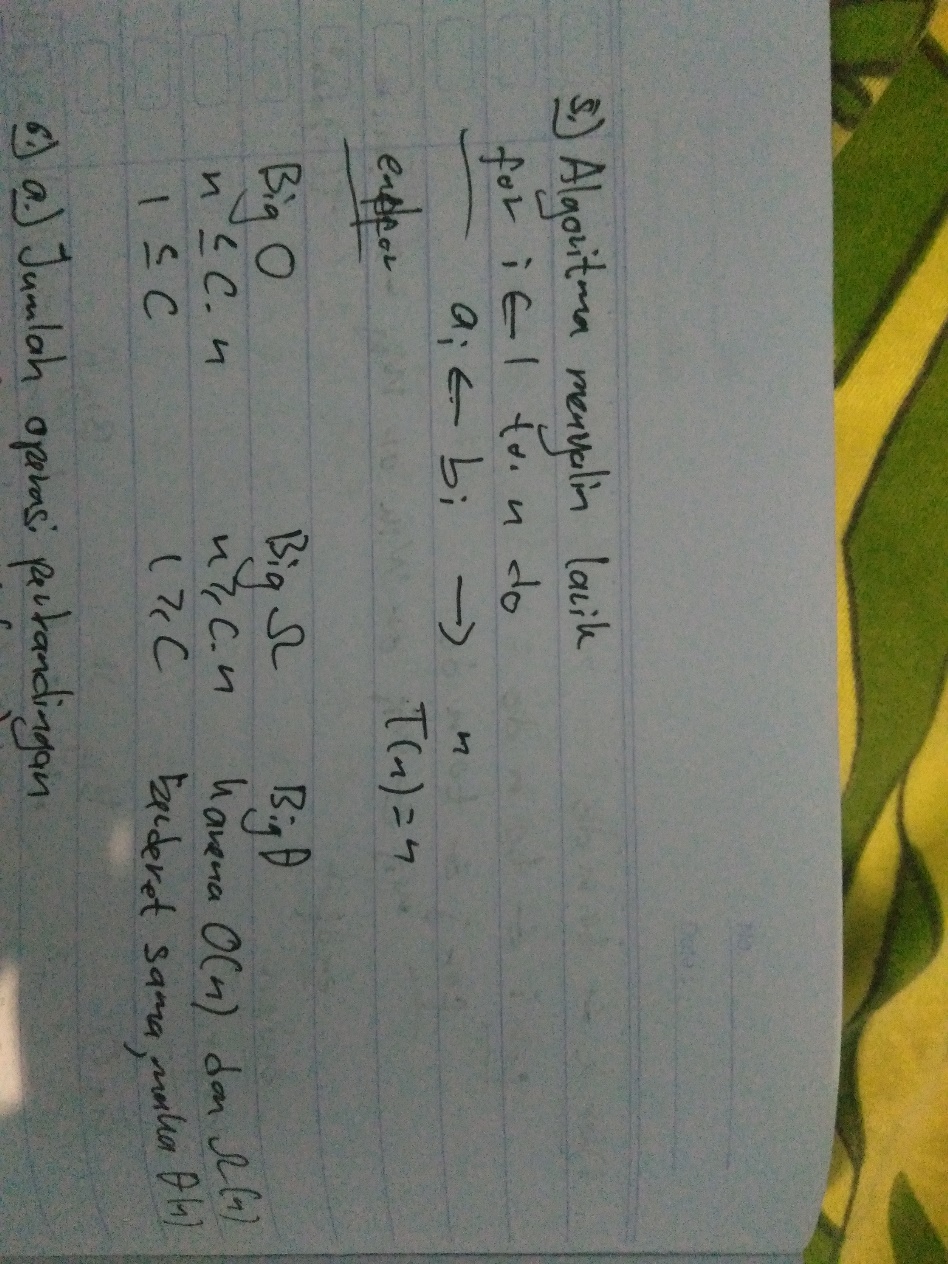
endfor



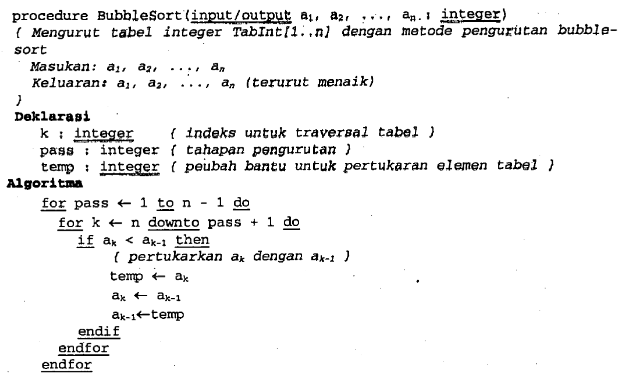
1. Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran . Berapa kompleksitas waktunya ? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big-Ω, dan Big-Θ?



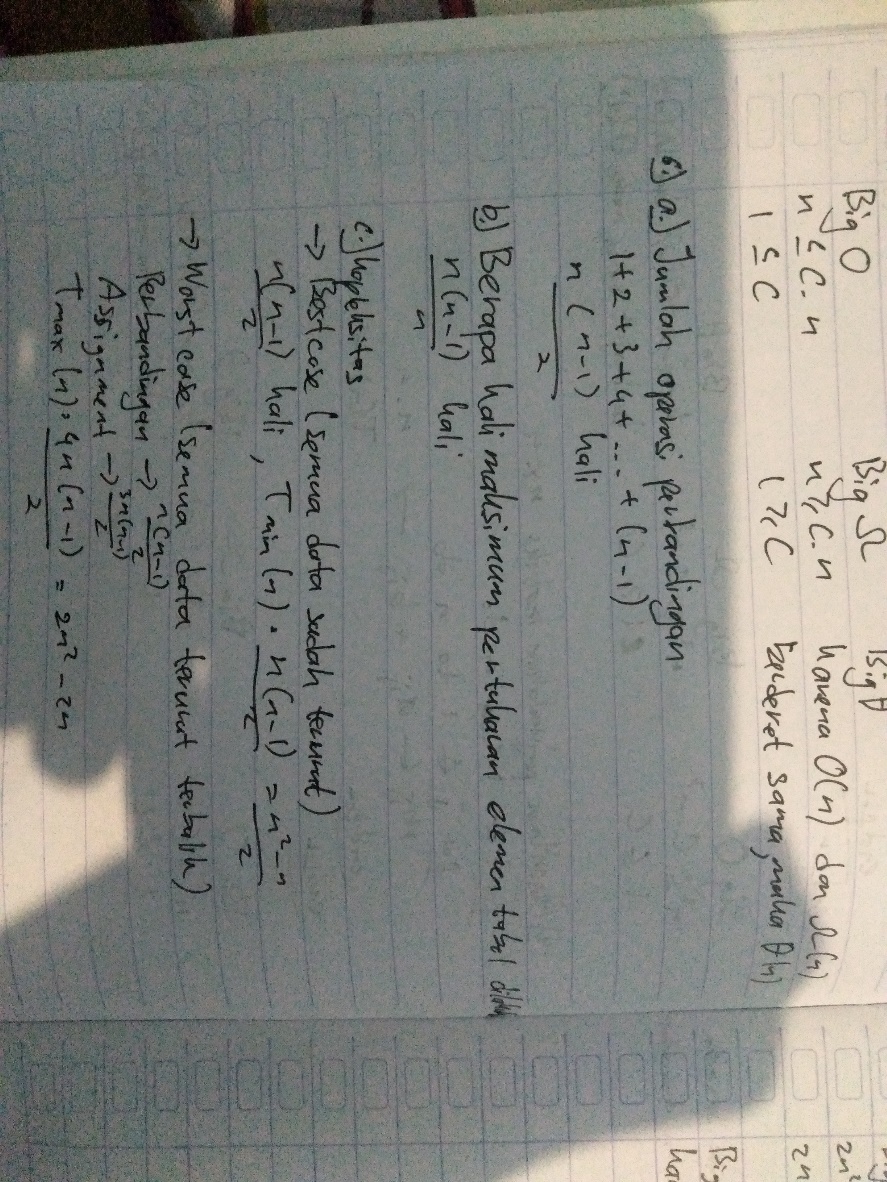
1. Tulislah algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah elemen. Berapa kompleksitas waktunya ? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big-Ω, dan Big-Θ?

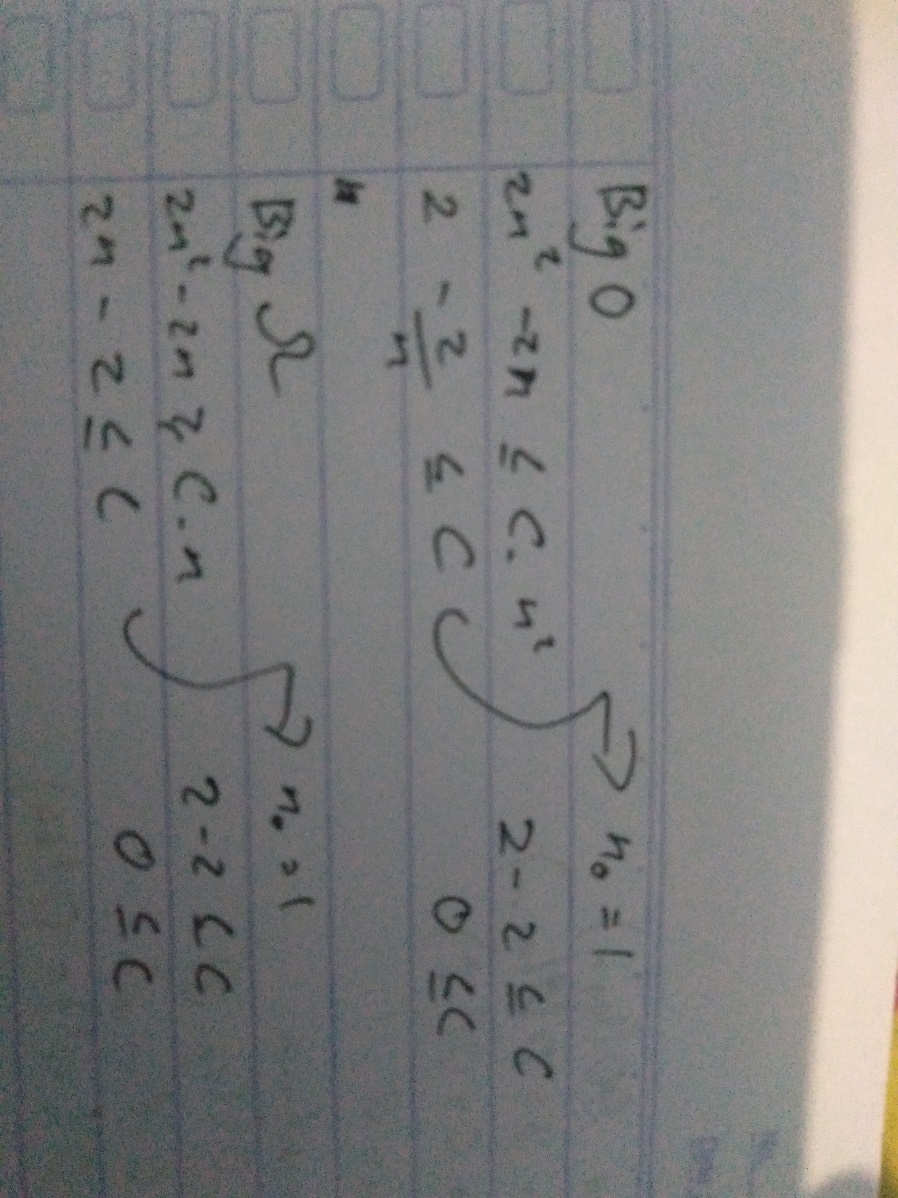


1. Diberikan algoritma Bubble Sort sebagai berikut:



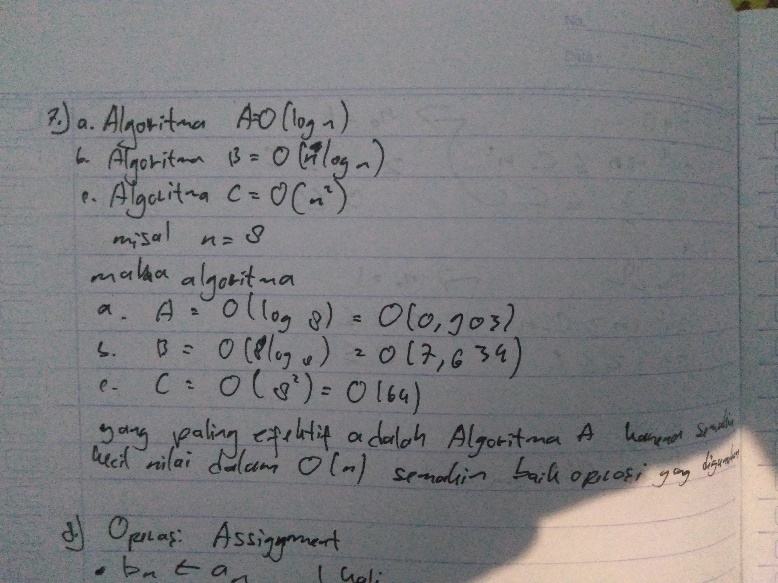
* 1. Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
  2. Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
  3. Hitung kompleksitas waktu asimptotik (Big-O, Big-Ω, dan Big-Θ) dari algoritma Bubble Sort tersebut!



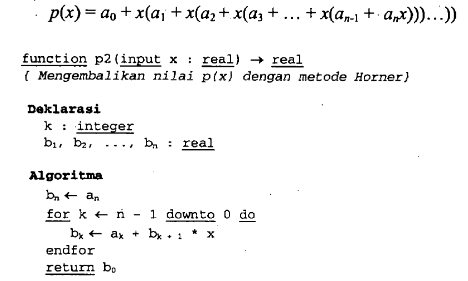


1. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:
   1. Algoritma A mempunyai kompleksitas waktu O(log N)
   2. Algoritma B mempunyai kompleksitas waktu O(N log N)
   3. Algoritma C mempunyai kompleksitas waktu O(N2)

Untuk problem X dengan ukuran N=8, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?



1. Algoritma mengevaluasi polinom yang lebih baik dapat dibuat dengan metode Horner berikut:



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?

