**LAPORAN PRAKTIKUM 3**

**ANALISIS ALGORITMA**

****

**RIZKY ANUGERAH**

**140810180049**

**KELAS A**

Fakultas Matematika dan Ilmu Pengetahuan Alam

Program Studi Teknik Informatika

Universitas Padjadjaran

2020

**PENDAHULUAN**

Minggu lalu kita sudah mempelajari menghitung kompleksitas waktu T(n) untuk semua operasi

yang ada pada suatu algoritma. Idealnya, kita memang harus menghitung semua operasi tersebut.

Namun, untuk alasan praktis, kita cukup menghitung operasi abstrak yang **mendasari suatu**

**algoritma**, dan memisahkan analisisnya dari implementasi. Contoh pada algoritma searching,

operasi abstrak yang mendasarinya adalah operasi perbandingan elemen x dengan elemen-elemen

dalam larik. Dengan menghitung berapa perbandingan untuk tiap-tiap elemen nilai n sehingga kita

dapat memperoleh **efisiensi relative** dari algoritma tersebut. Setelah mengetahui T(n) kita dapat

menentukan **kompleksitas waktu asimptotik** yang dinyatakan dalam notasi Big-O, Big-Ω, Big-Θ,

dan little-ω.

Setelah mengenal macam-macam kompleksitas waktu algoritma (best case, worst case, dan

average case), dalam analisis algoritma kita selalu mengutamakan perhitungan **worst case** dengan

alasan sebagai berikut:

• Worst-case running time merupakan *upper bound* (batas atas) dari running time untuk

input apapun. Hal ini memberikan jaminan bahwa algoritma yang kita jalankan tidak akan

lebih lama lagi dari ***worst-case***

• Untuk beberapa algoritma, ***worst-case*** cukup sering terjadi. Dalam beberapa aplikasi

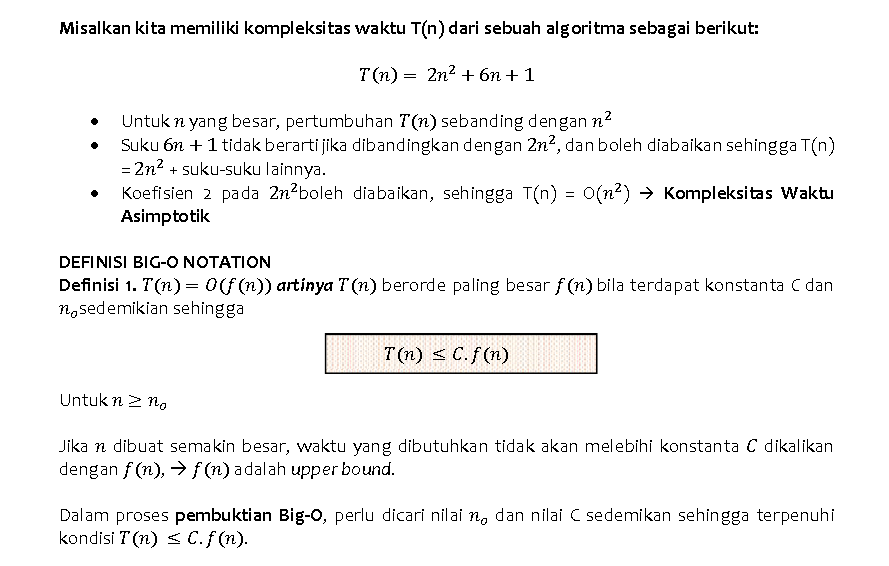
pencarian, pencarian info yang tidak ada mungkin sering dilakukan.

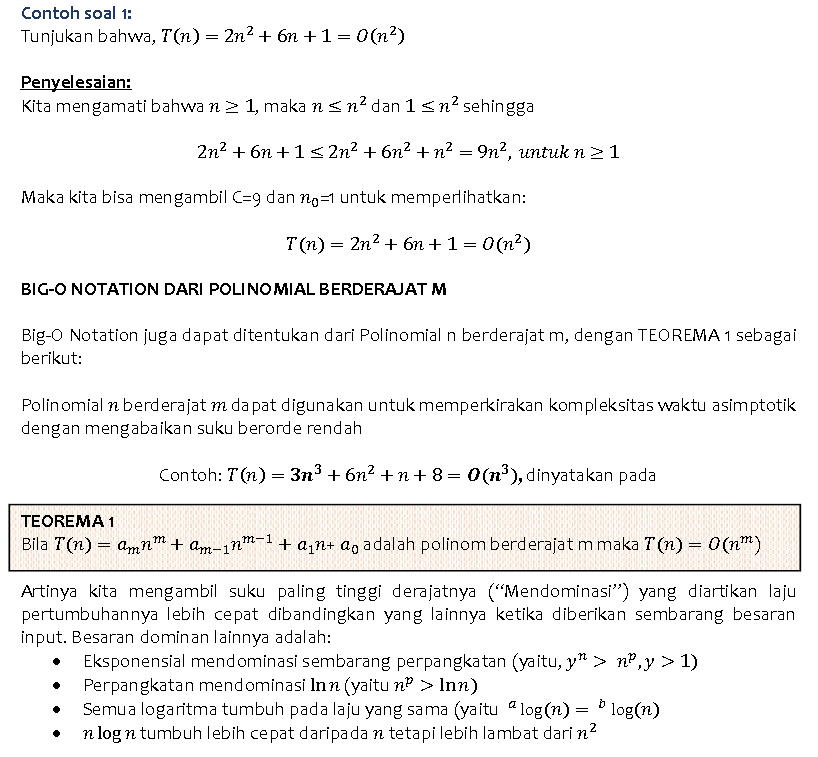
• Pada kasus ***average-case*** umumnya lebih sering seperti ***worst-case.*** *Contoh:* misalkan kita

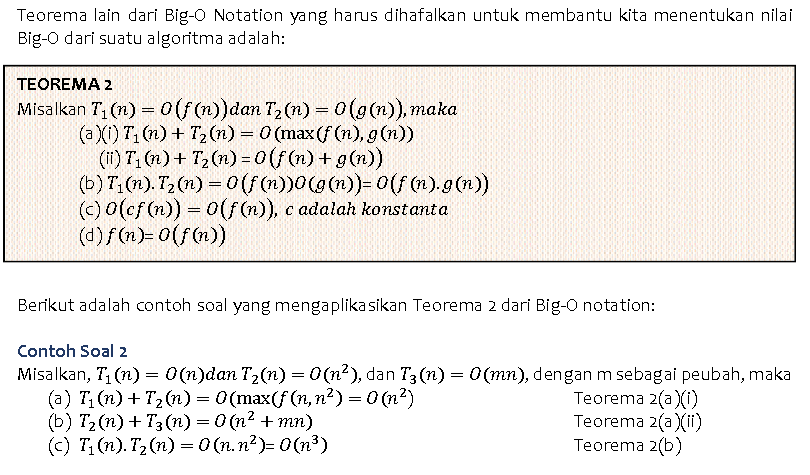
secara random memilih n angka dan mengimplementasikan insertion sort, ***average-case =***

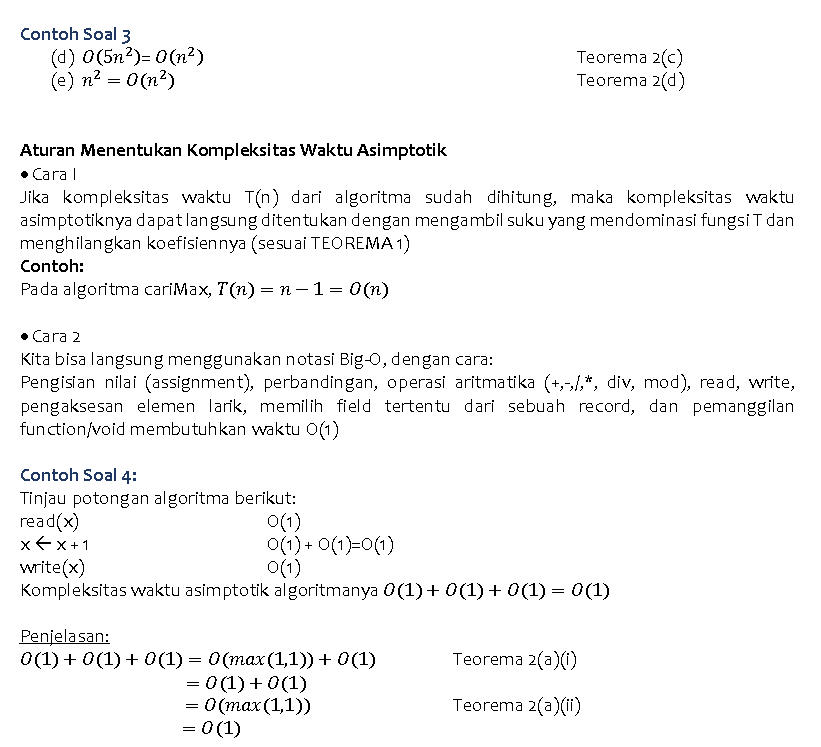
***worst-case*** yaitu fungsi kuadratik dari n.

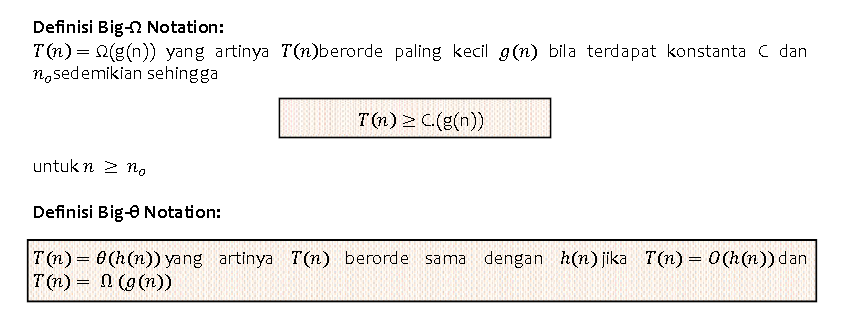
Perhitungan worst case (*upper bound*) dalam kompleksitas waktu asimptotik dapat menggunakan **Big-O Notation. Perhatikan pembentukan Big-O Notation berikut!**

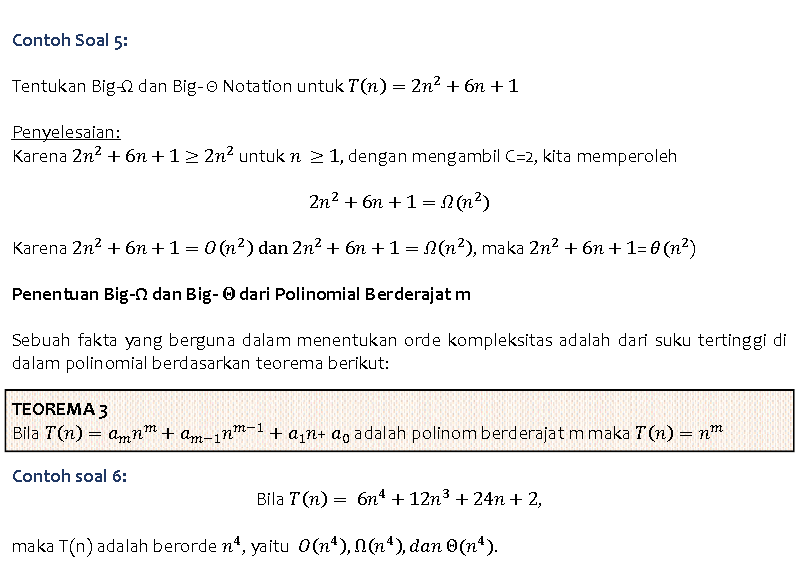




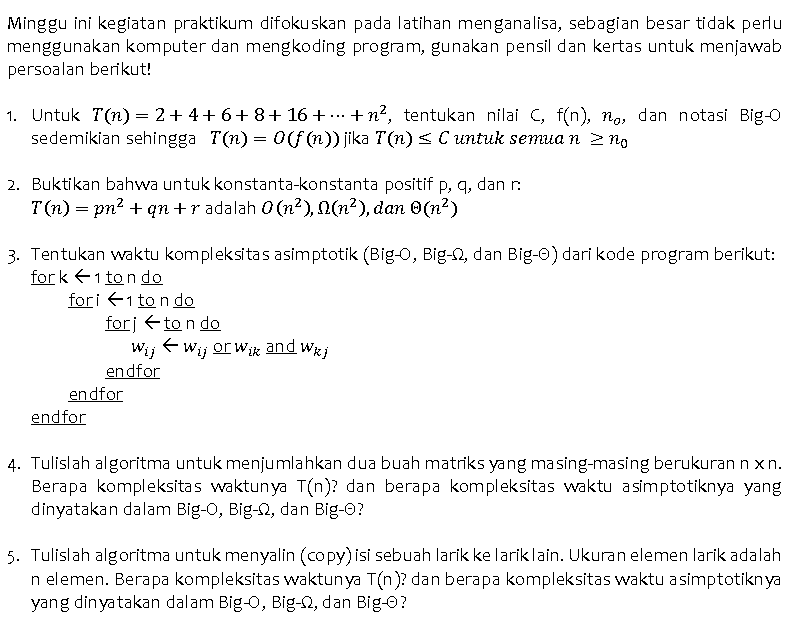




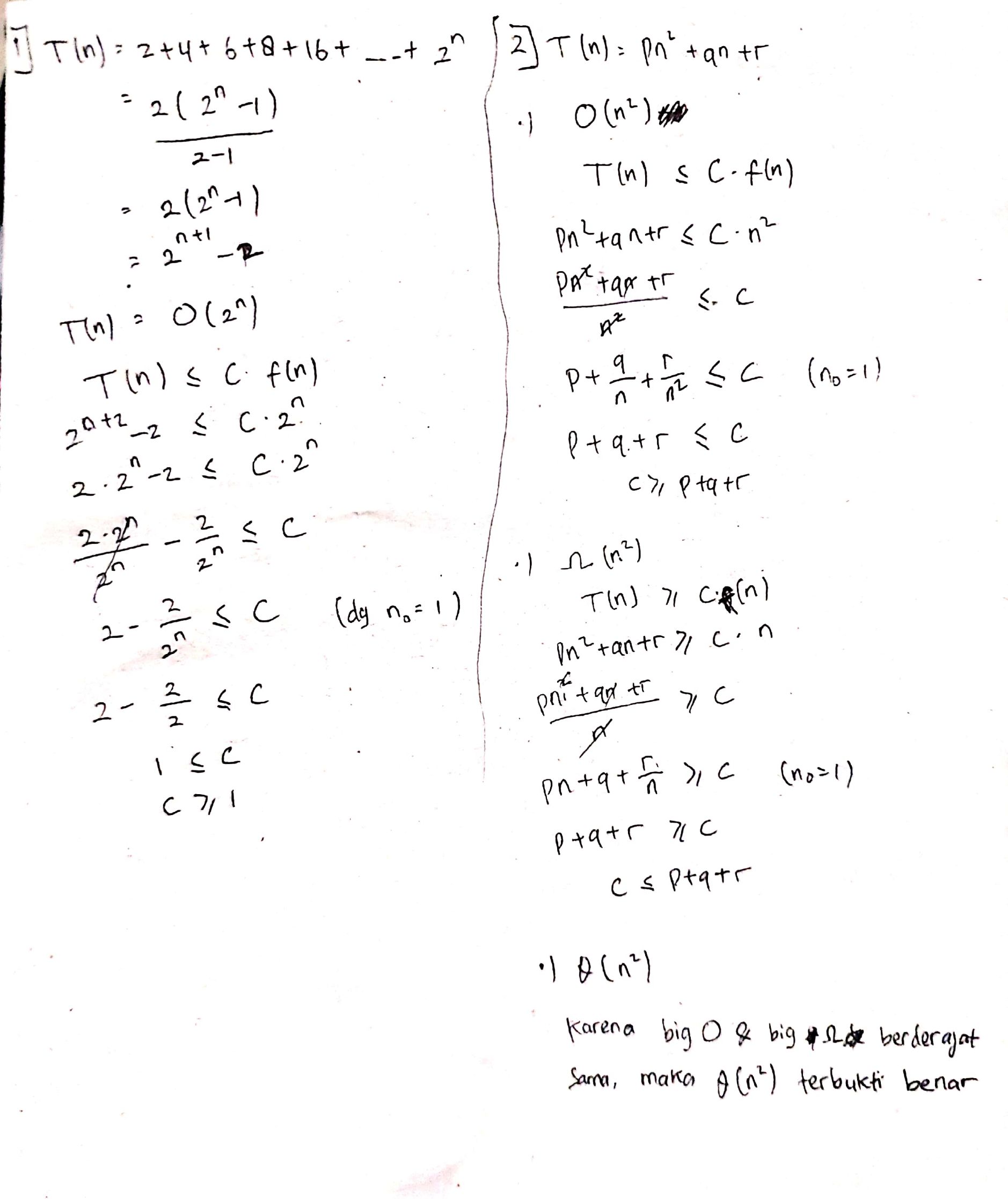


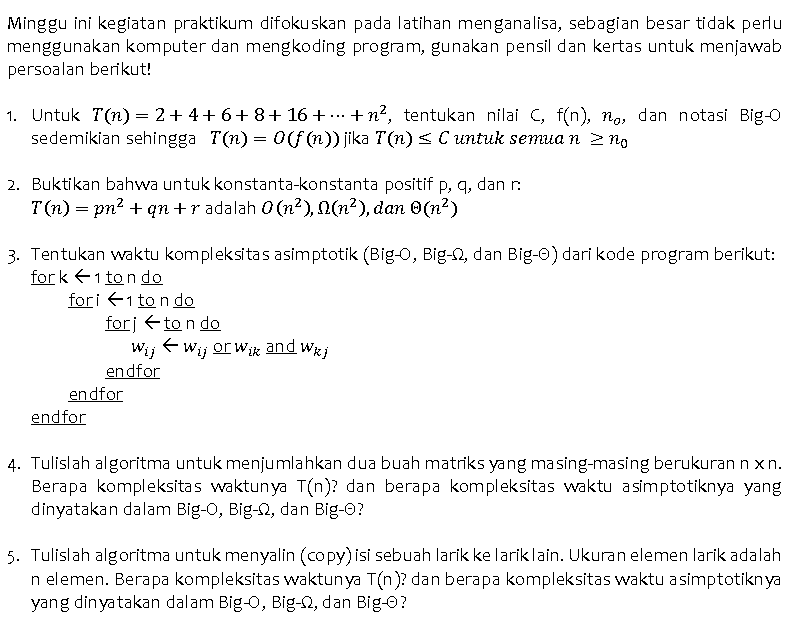


**LATIHAN**

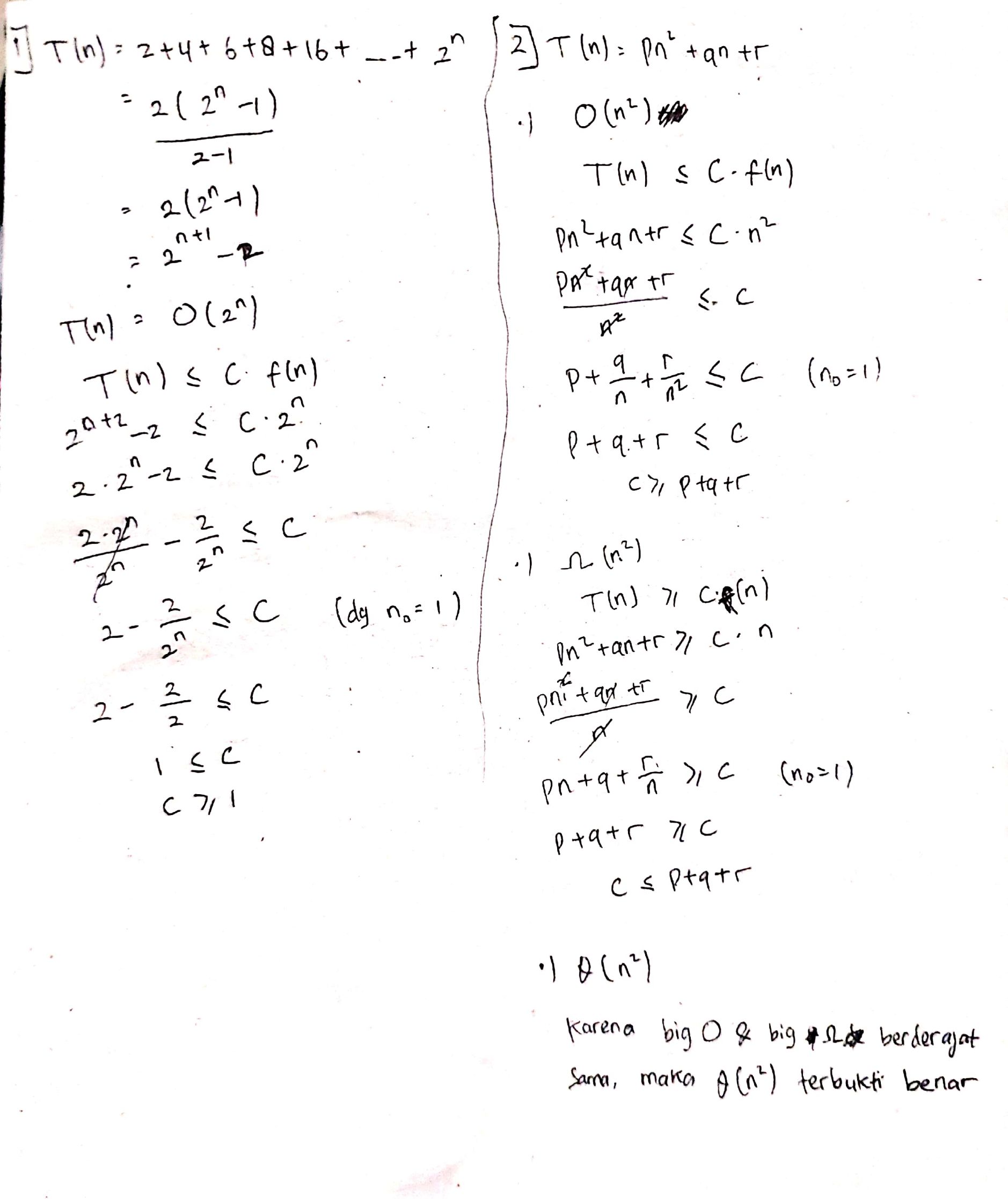


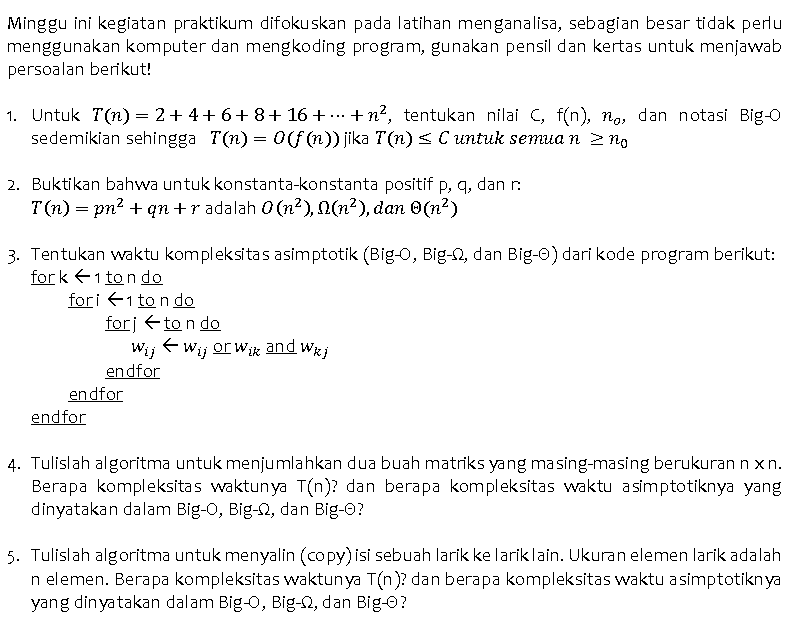
Jawab:



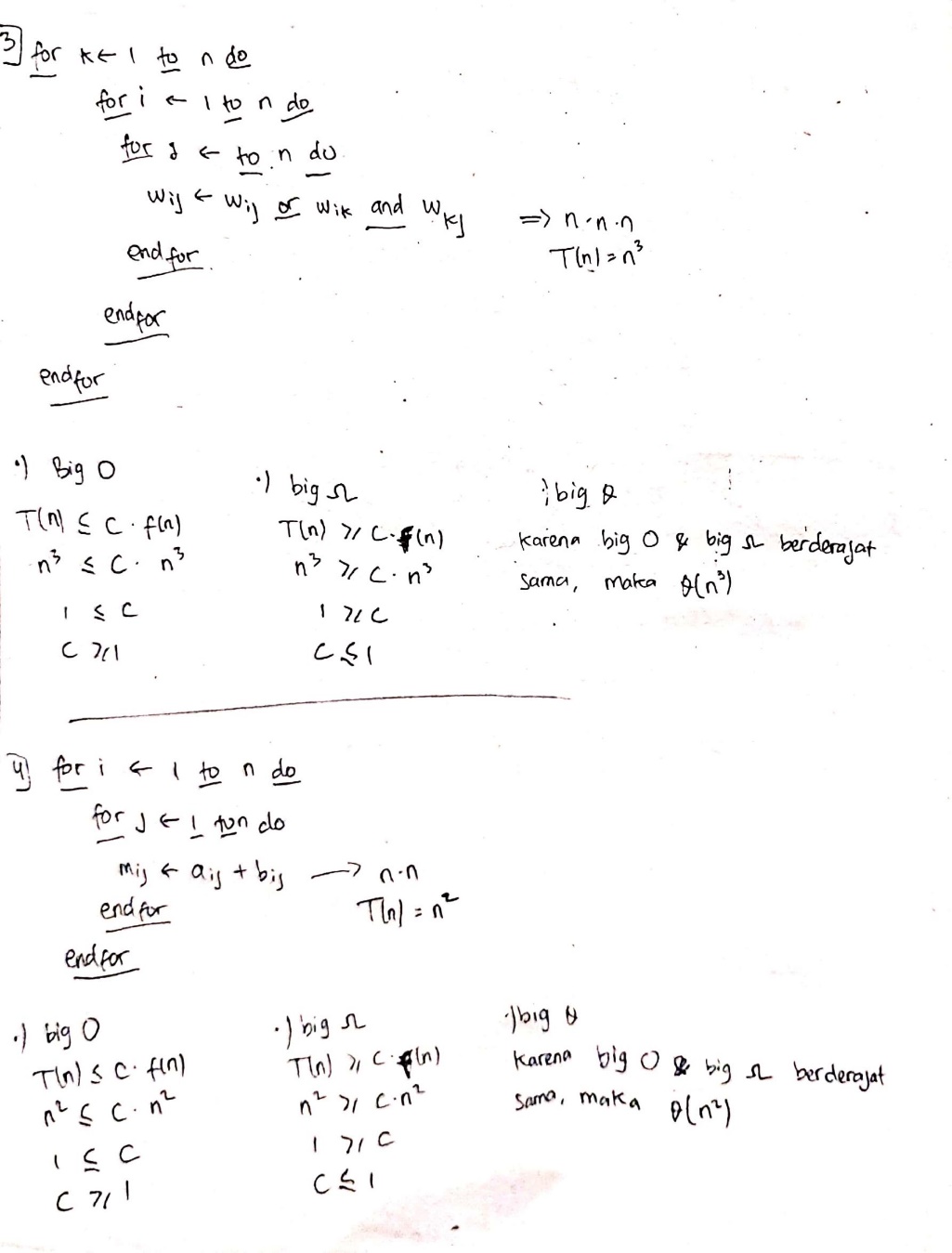


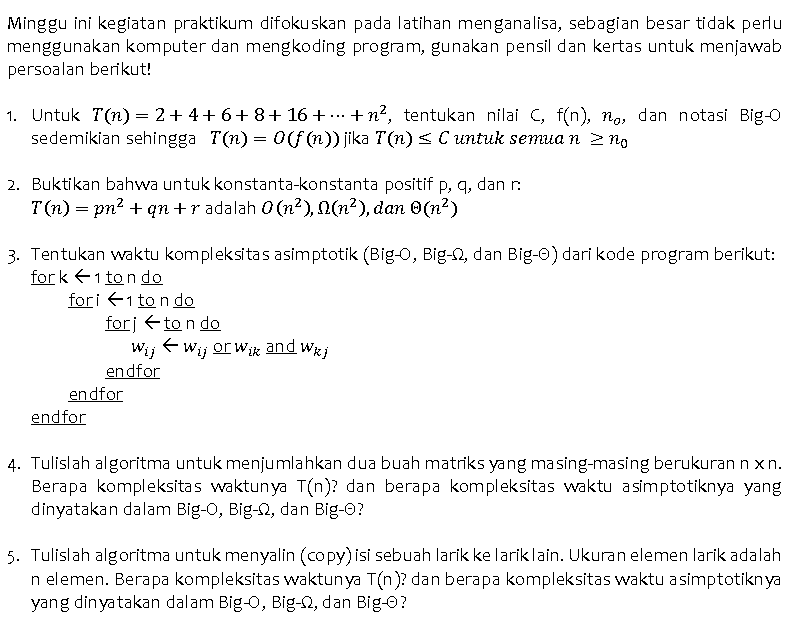
Jawab:



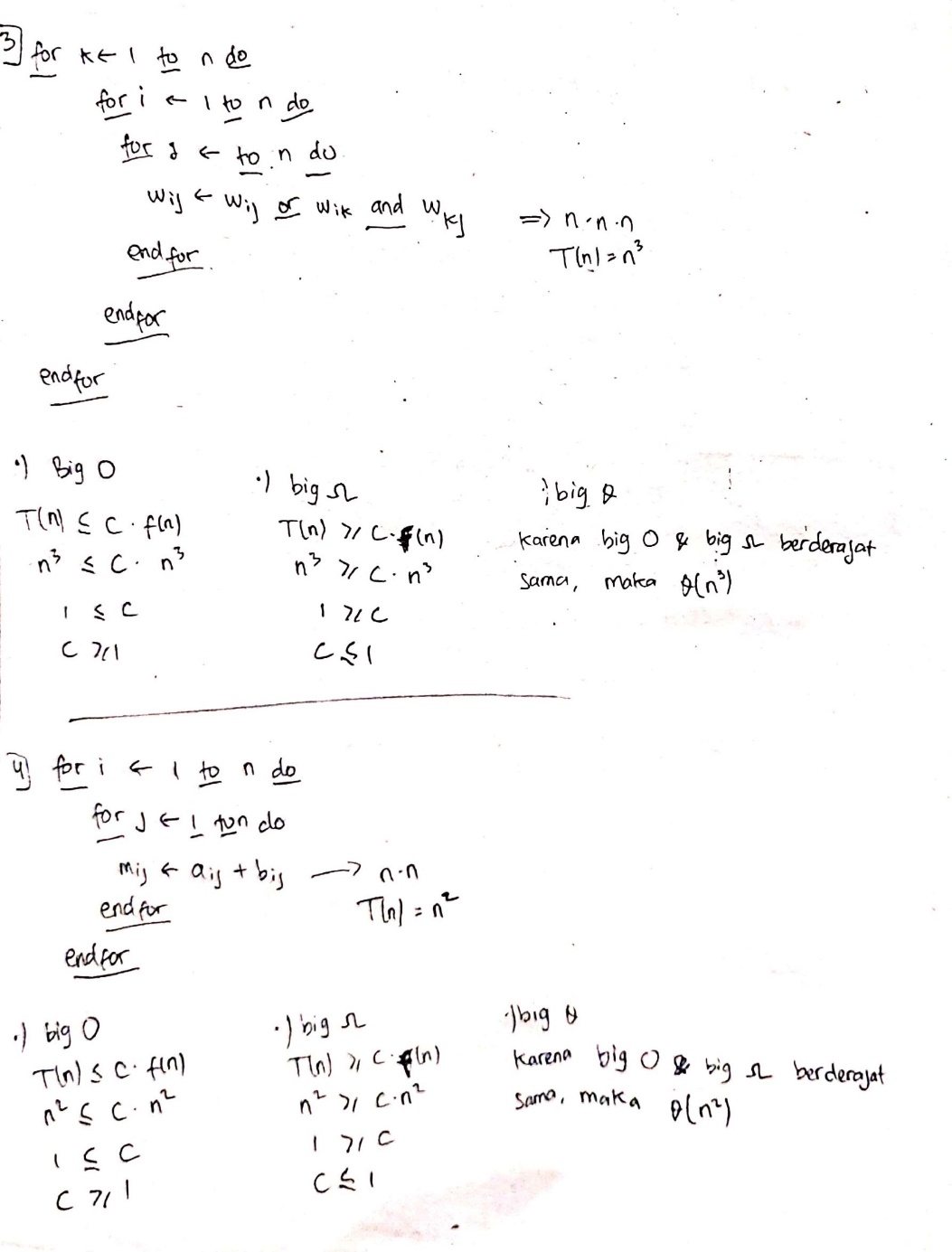


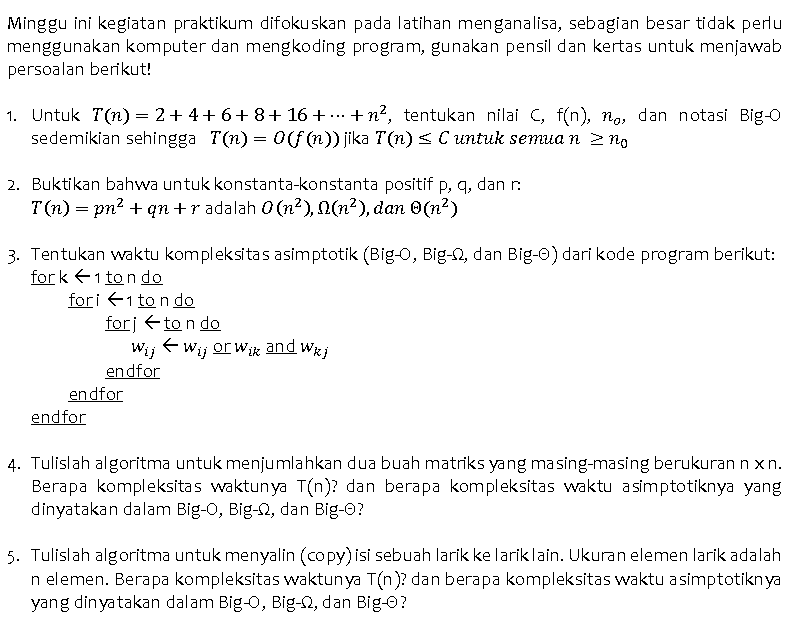
Jawab:



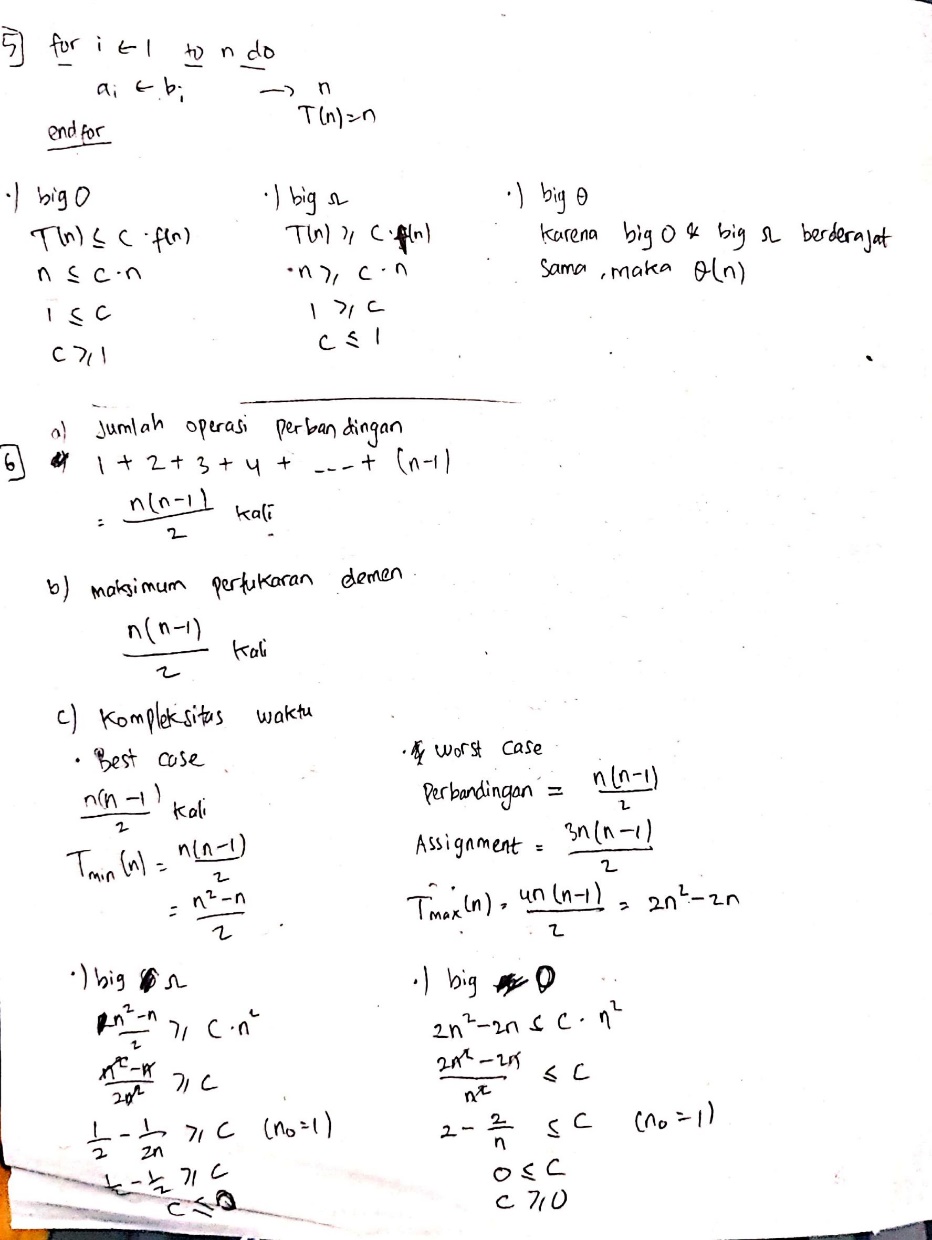


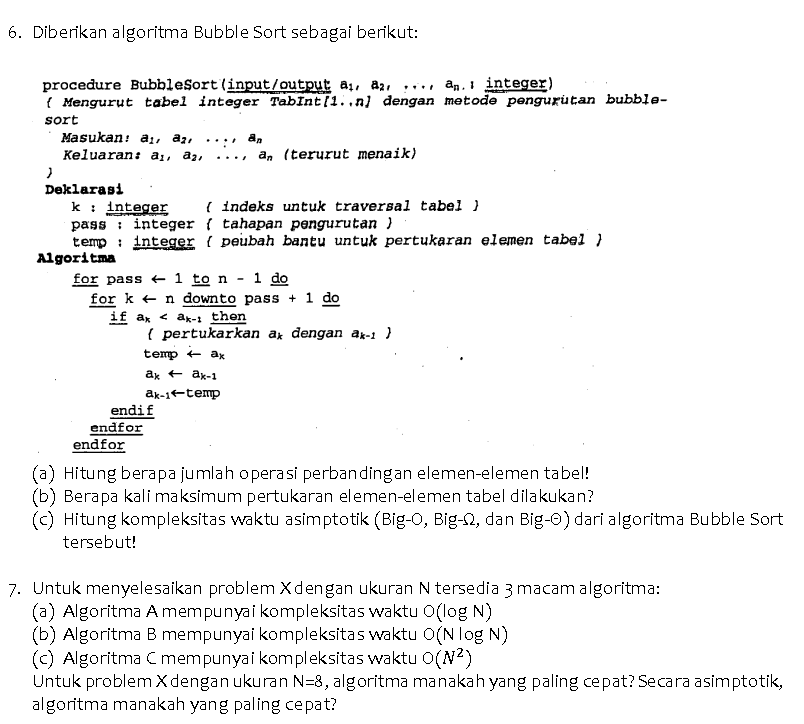
Jawab:



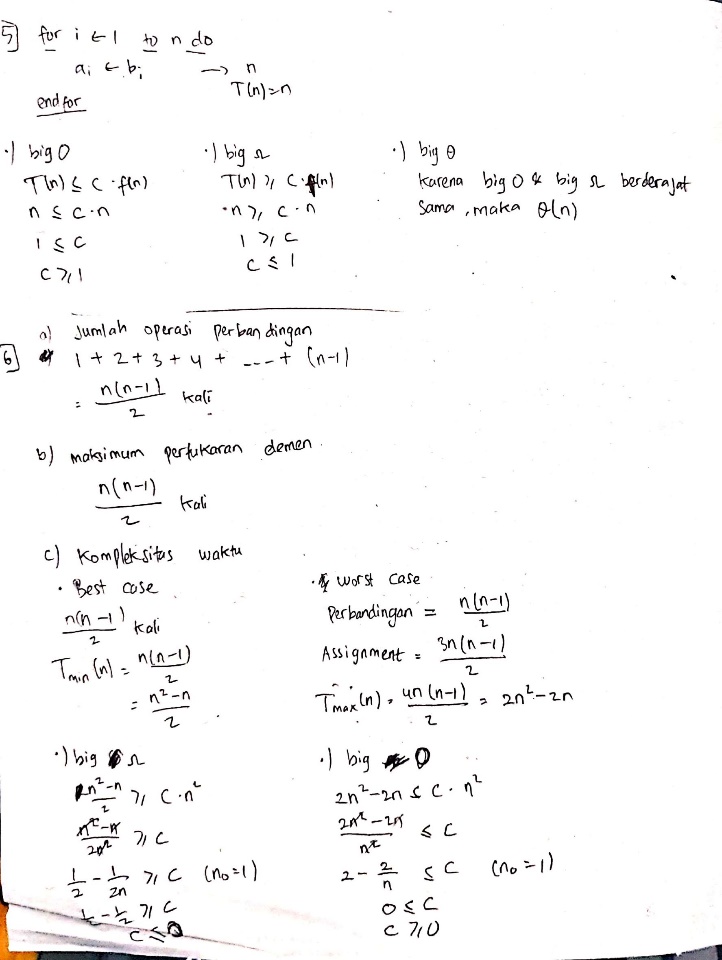


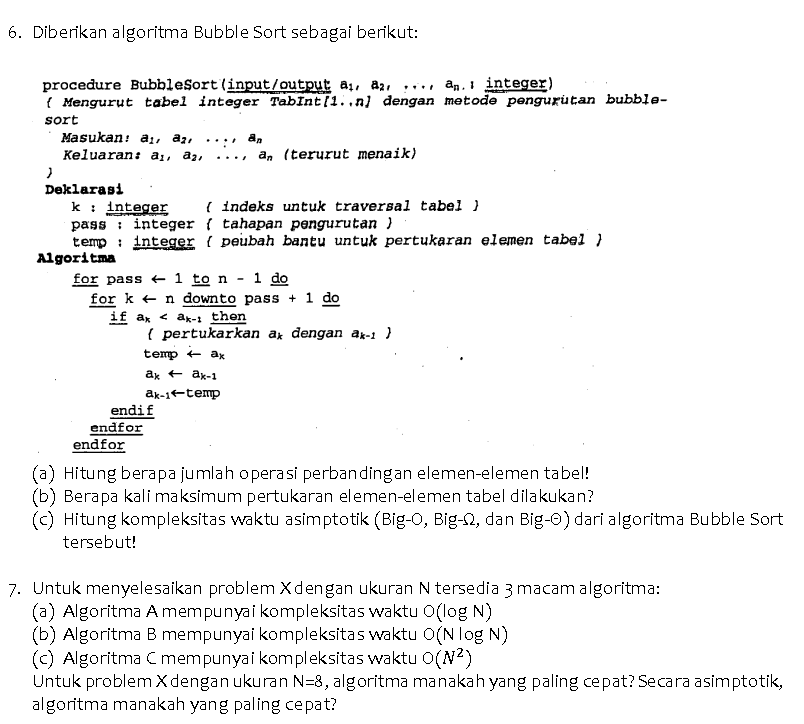
Jawab:



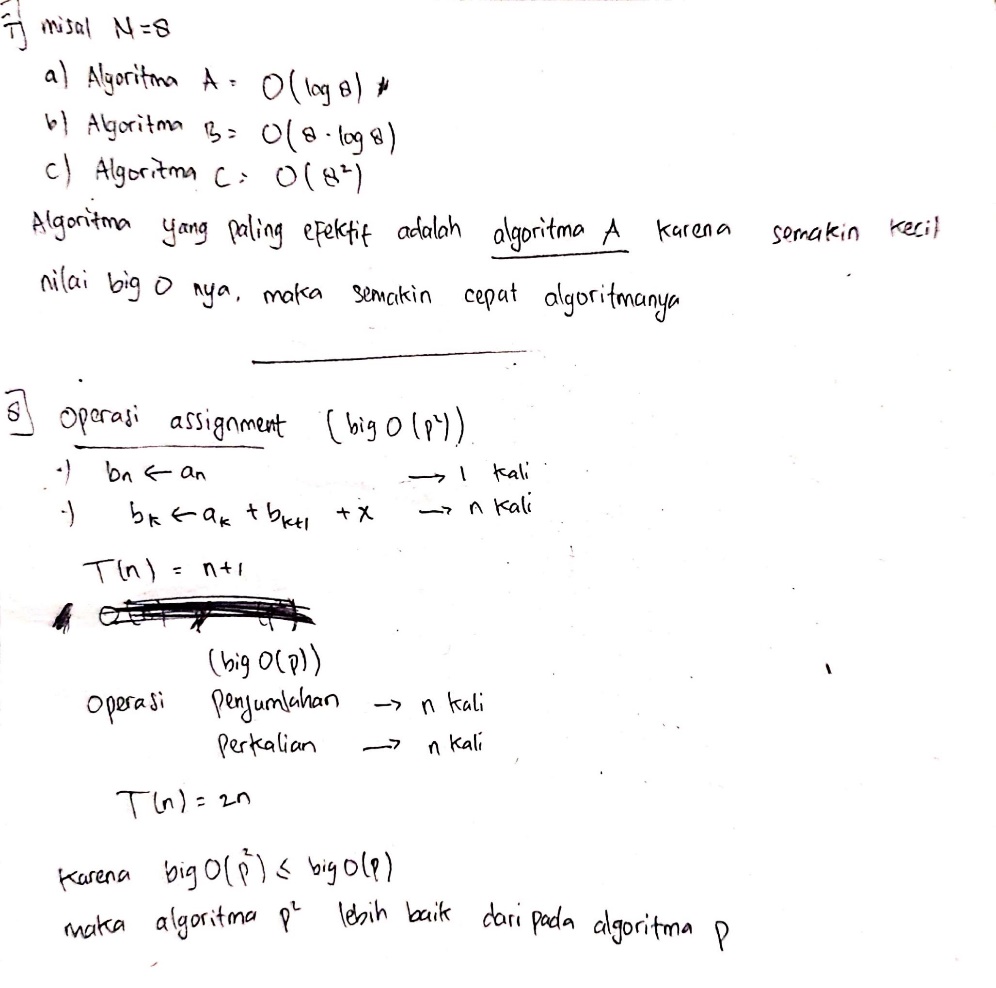


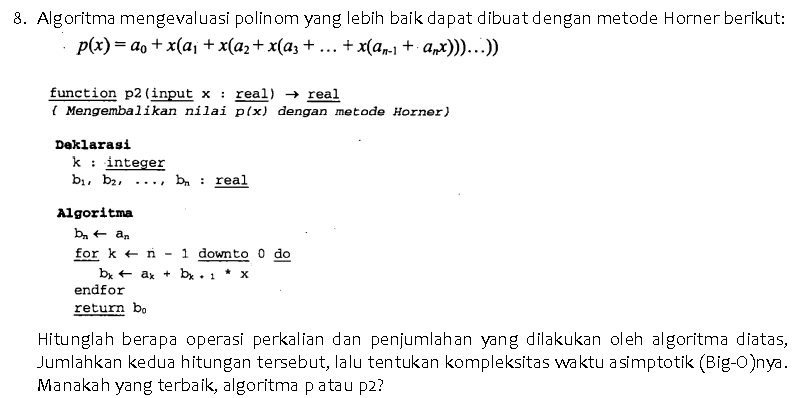
Jawab:





Jawab:





Jawab:

