CLO 2

ASSIGNMENT 5

SOAL 1

$T(n) = 2^n$.
Ukuran input 32 dapat dieksekusi dalam 1 menit $\Rightarrow$ komputer dapat mengeksekusi 2^{32} operasi dasar dalam 1 menit.
Jika ukuran input 64, maka komputer dapat mengeksekusi algoritma tersebut dalam $\frac{2^{64}}{2^{32}} = 2^{32}$ menit, yang jelas jauh melebihi batas waktu, yaitu kisaran 1 menit. Oleh karenanya, **dibutuhkan** komputer yang baru untuk mengeksekusi algoritma yang sama dengan ukuran input 64 jika ingin diselesaikan dalam waktu kisaran 1 menit.

SOAL 2

$T(n) = \log_2 n$.
Ukuran input 32 dapat dieksekusi dalam 1 menit $\Rightarrow$ komputer dapat mengeksekusi $log_2 32 = 5$ operasi dasar dalam 1 menit.
Jika ukuran input 64, maka komputer dapat mengeksekusi algoritma tersebut dalam $\frac{\log_2 64}{5} = 1,2$ menit, yang cukup dekat dengan batas waktu 1 menit. Oleh karenanya, **tidak dibutuhkan** komputer yang baru untuk mengeksekusi algoritma yang sama dengan ukuran input 64 jika ingin diselesaikan dalam waktu kisaran 1 menit.

SOAL 3

$T(n) = n^3$.
Ukuran input 32 dapat dieksekusi dalam 1 menit $\Rightarrow$ komputer dapat mengeksekusi 32^3 operasi dasar dalam 1 menit.
Jika ukuran input 64, maka komputer dapat mengeksekusi algoritma tersebut dalam $\frac{64^3}{32^3} = 2^3 = 8$ menit, yang jelas melebihi batas waktu, yaitu kisaran 1 menit. Oleh karenanya, **dibutuhkan** komputer yang baru untuk mengeksekusi algoritma yang sama dengan ukuran input 64 jika ingin diselesaikan dalam waktu kisaran 1 menit.

SOAL 4

$T(n) = n \log_2 n$.
Ukuran input 32 dapat dieksekusi dalam 1 menit $\Rightarrow$ komputer dapat mengeksekusi $32 log_2 32 = 160$ operasi dasar dalam 1 menit.
Jika ukuran input 64, maka komputer dapat mengeksekusi algoritma tersebut dalam $\frac{64 \log_2 64}{160} = 2,4$ menit, yang cukup dekat dengan batas waktu 2 menit. Oleh karenanya, **tidak dibutuhkan** komputer yang baru untuk mengeksekusi algoritma yang sama dengan ukuran input 64 jika ingin diselesaikan dalam waktu kisaran 2 menit.

ASSIGNMENT 6

SOAL 1

Perhatikanlah algoritma berikut:

```
Function Transpose_Matrix(A, B: array[1..n,1..n] of integer)→ Boolean

Kamus
i,j: integer
stop: Boolean

Algoritma

stop ← False
i ← 1; j ← 1
while (i <= n) and not stop do
   while (j <= n) and not stop do
      if A[i,j] = B[j,i] then
         j ← j + 1
      else
         stop ← True
   i ← i + 1
return stop
```

a) Tentukan ukuran input pada algoritma tersebut: dimensi matriks, $n \; x \; n$

b) Tentukan operasi dasar pada algoritma tersebut: perbandingan $A[1,i] \; = \; B[j,i]$

c) Tentukan apakah ada kasus terbaik, terburuk,? Jelaskan!

Kasus **terbaik** jika $A[1,1] = B[1,1]$, maka $T(n) = 1$

Kasus **terburuk**:

$$T(n) = \sum_{i=1}^{n}\sum_{j=1}^{n} 1 = \sum_{i=1}^{n} n - 1 + 1 = n\sum_{i=1}^{n} 1 = n^2 \epsilon \theta(n^2)$$

SOAL 2

Perhatikanlah algoritma berikut:

```
Function Max_Subarray(A: array[1..n] of integer)→ integer

Kamus
sum, max: integer

Algoritma

max ← 0
for i ← 1 to n do
   sum ← 0
   for j ← i to n do
      sum ← sum + A[j]
      if sum > max then
         max ← sum
return max
```

a) Tentukan ukuran input pada algoritma tersebut: ukuran array, n

b) Tentukan operasi dasar pada algoritma tersebut: perbandingan $sum \; > \; max$

c) Tentukan apakah ada kasus terbaik, terburuk? Jelaskan!

Tidak ada kasus terbaik/terburuk.

Every case time complexity:

$$T(n) = \sum_{i=1}^{n}\sum_{j=i}^{n} 1 = \sum_{i=1}^{n} n - i + 1 = \frac{3}{2}n^2 + \frac{3}{2}n \in \theta(n^2)$$

SOAL 3

Perhatikanlah algoritma berikut:

```
Function Transpose_Matrix(A: array[1..n] of integer)→ Boolean


Algoritma

for i ← n to 2 do
   for i ← i-1 to 1 do
      if A[i] = B[j] then
         return False
return True
```

a) Tentukan ukuran input pada algoritma tersebut: ukuran array, n

b) Tentukan operasi dasar pada algoritma tersebut: perbandingan $A[i] = B[j]$

c) Tentukan apakah ada kasus terbaik, terburuk? Jelaskan!

Kasus **terbaik** jika $A[n] = B[n-1]$, maka $T(n) = 1$

Kasus **terburuk**:

$$T(n) = \sum_{i=2}^{n}\sum_{j=1}^{i-1} 1 = \sum_{i=2}^{n} i - 1 - 1 + 1 = n\sum_{i=2}^{n} i - 1 = \frac{n(n+1)}{2} - 2 - (n-2+1) = \frac{n(n+1)}{2} - n - 1 = \frac{1}{2}n^2 - \frac{1}{2}n - 1 \in \theta(n^2)$$

ASSIGNMENT 8

SOAL 1

Lakukan analisis untuk mendapatkan *time efficiency class* (kompleksitas) dari *Element Uniqueness Problem* berikut yang memeriksa apakah semua elemen dalam sebuah kelompok semuanya berbeda.

```
Algorithm UniqueElement ( A[ 1...n ] ):
i ← 1
j ← 1
while i < n do
        j ← i + 1
        while j <= n do
                if A[ i ] = A[ j ] then
                        return FALSE
                end if
        end while
end while
```

SOLUSI

$$C_{worst}(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} = \sum_{i=1}^{n-1} [n-(i+1)+1] = \sum_{i=1}^{n-1} (n-i)$$
$$= (n-1)+(n-2)+\ldots+1 = \frac{(n-1)n}{2} \in O(n^2)$$

SOAL 2

Lakukan analisis untuk mendapatkan *time efficiency class* (kompleksitas) dari algoritma *Power* berikut yang menghitung bilangan x jika dipangkatkan dengan bilangan y.

```
Algorithm Power ( x, y ):
count ← 1
result ← 1
while count <= y do
        result ← result * x
        count ← count + 1

end while
return result
```

SOLUSI

$$C(n) = \sum_{count=1}^{y} 1 = y \in O(y)$$

SOAL 3

```
Algorithm Selection-Sort ( A[ 1...n ] )
for i ← 1 to n-1 do
        minJ ← i
        minX ← A[ i ]
                for j ← i + 1 to n do
                        if A[ j ] < minX then
                                minJ ← j
                                minX ← A[ j ]
                        end if
                end for


        A[ minJ ] ← A [ i ]
        A[ i ] ← minX
 end for
```

SOLUSI

Untuk setiap i dari 1 hingga $n-1$, ada 1 kali pertukaran nilai dan $n-i$ perbandingan, sehingga:

Total pertukaran: $E(n) = n-1$

Total perbandingan: $C(n) = (n-1)+(n-2)+\ldots+1 = \frac{(n-1)n}{2}$

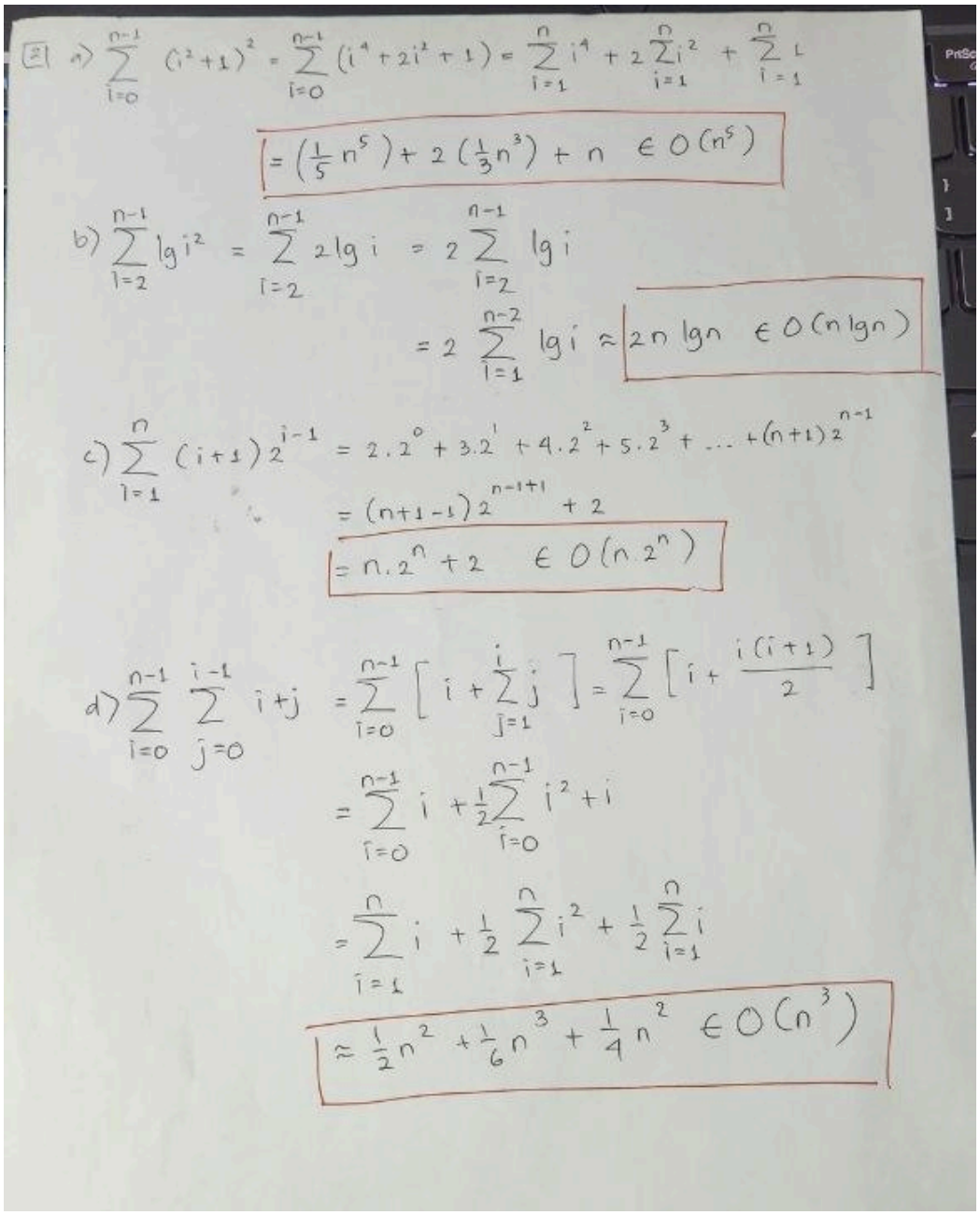
$T(n) = E(n) + C(n)$

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

SOAL 4

Tentukan kelas kompleksitas algoritma yang memiliki $C(n)$ di bawah ini!

1. $C(n) = \sum_{i=0}^{n-1}(i^2+1)^2$
2. $C(n) = \sum_{i=2}^{n-1} \log i^2$
3. $C(n) = \sum_{i=1}^{n}(i+1).2^{i-1}$
4. $C(n) = \sum_{i=0}^{n-1}\sum_{j=0}^{i-1}(i+j)$



SOAL 5

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= m; j++) {
        sum++;
    }
}
```

SOLUSI:

$T(n) = \sum_{i=1}^{n}\sum_{j=1}^{m} 1 = \sum_{i=1}^{n} m - 1 + 1 = \sum_{i=1}^{n} m = m\sum_{i=1}^{n} 1 = m(n-1+1) = m.n \in \theta(m.n)$

SOAL 6

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i < n; i++) {
    sum++;
}
```

SOLUSI:

$$T(n) = \sum_{i=1}^{n-1} 1 = (n-1) - 1 + 1 = n - 1 \in \theta(n)$$

SOAL 7

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i < n; i=i+2) {
    sum++;
}
```

SOLUSI:

$$T(n) = \frac{1}{2}\sum_{i=1}^{n-1} 1 = \frac{1}{2}(n - 1 - 1 + 1) = \frac{1}{2}(n-1) \in \theta(n)$$

SOAL 8

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n/2; j++) {
        sum++;
    }
}
```

SOLUSI:

$$T(n) = \sum_{i=1}^{n}\sum_{j=1}^{n/2} 1 = \sum_{i=1}^{n}\frac{n}{2} - 1 + 1 = \sum_{i=1}^{n}\frac{n}{2} = \frac{n}{2}\sum_{i=1}^{n} 1 = \frac{n}{2}(n) = \frac{1}{2}n^2 \in \theta(n^2)$$

SOAL 9

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i < n; i = i*2) {
    sum++;
}
```

SOLUSI:

Perhatikan nilai i pada tiap iterasi. Andaikan $i = 2^k$ untuk suatu $k \in \mathbb{Z}^+$

$$i_0 = 2^k$$
$$i_1 = 2^1$$
$$i_2 = 2^2$$
$$i_3 = 2^3$$
$$i_t = n = 2^t$$

Sehingga, $t = k$. Maka, $n = 2^t = 2^k \longrightarrow k = \log_2 n$

Jadi, operasi penjumlahan dilakukan sebanyak $\log_2 n \longrightarrow O(\log_2 n)$

SOAL 10

Tentukan kompleksitas asimtotik dari algoritma berikut

```
int sum = 0;
for (int i = 1; i < n; i=i+2) {
    for (int j = 1; j < i; j++) {
        sum++;
    }
}
```

SOLUSI:

$$T(n) = \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{i-1} 1 = \frac{1}{2}\sum_{i=1}^{n-1} i - 1 - 1 + 1 = \frac{1}{2}\sum_{i=1}^{n-1} i - 1 = \frac{1}{2}\sum_{i=1}^{n-1} i - \frac{1}{2}\sum_{i=1}^{n-1} 1 = \frac{1}{2}\frac{(n-1)n}{2} - \frac{1}{2}(n-1)$$
$$= \frac{1}{4}n^2 - \frac{3}{4}n + 1 \in \theta(n^2)$$