Nama : Okka Riswana
NPM : 140810180032

Worksheet Modul 3

(1.) $T(n) = 2+4+6+8+\dots+n^2$

$= \sum_{i=1}^{n^2} 2i = n^2(n^2+1)$

$= n^4+n^2$

Dengan $n \geq 1$, maka $n^2 \leq n^4$ sehingga

$n^4+n^2 \leq n^4+n^4 = 2n^4$, $n \geq 1$

$\therefore T(n) = 2+4+6+8+\dots+n^2 = O(n^4)$ dengan $c \geq 2$, $n_0 = 1$, $f(n) = 2n^4$

(2.) $T(n) = pn^2+qn+r$

$O \rightarrow pn^2+qn+r \leq (p+q+r)n^2$, $n \geq 1$

dengan $c \geq (p+q+r)$ dan $n_0 = 1$, maka
$T(n) = O(n^2)$

$\Omega \rightarrow pn^2+qn+r \geq pn^2$, $n \geq 0$

dengan $c \geq p$ dan $n_0 = 1$, maka
$T(n) = \Omega(n^2)$

$\Theta \rightarrow$ karena $O(n^2)$ dan $\Omega(n^2)$, maka
$T(n) = \Theta(n^2)$

(3.) Jika operasi assignment kita anggap dalam waktu linear, maka

$T(n) = \sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} 1 = n^3$

$T(n) = O(n^3) = \Omega(n^3) = \Theta(n^3)$

(4.) General implementation

```
for i ← 1 : n
    for j ← 1 : n
        c[i][j] = a[i][j] + b[i][j]
```

Jika kita anggap operasi penambahan linear, maka

$T(n) = \sum_{i=1}^{n}\sum_{j=1}^{n} 1 = n^2$

$T(n) = O(n^2) = \Omega(n^2) = \Theta(n^2)$

(5.) General implementation

```
for i ← 1 : n
    b[i] = a[i]
```

Anggap assignment linear, maka

$T(n) = \sum_{i=1}^{n} 1 = n$

$O(n)$, $\Omega(n)$, $\Theta(n)$

(6.) a.) Perbandingan $= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} 1 = n(n-1)$

b.) Pada worst casenya, operasi swap akan sama dengan op. perbandingan :
$n(n-1)$

c.) $T(n) = n^2-n$

$O \rightarrow$ dengan $n \geq 0$ dan $c \geq 1$
$n^2-n \leq cn^2$
maka $T(n) = O(n^2)$

$\Omega \rightarrow$ dengan $n \geq$ dan $0 < c \leq 1$
$n^2-n \geq cn^2$
maka $T(n) = \Omega(n^2)$

$\Theta \rightarrow$ karena $O(n^2)$ dan $\Omega(n^2)$, maka
$T(n) = \Theta(n^2)$

(7.) Kita tidak bisa mengetahui mana yang lebih cepat untuk $n=8$ karena kita tidak tahu konstanta c yang dipakai.
Namun secara asimtotik algoritma (A) dengan $O(\log n)$ paling cepat

(8.) waktu asimtotik P_2

$$T(n) = n + n$$

$$\rightarrow O(n) \text{ dengan } c \geq 2,\ n_0 = 0$$

Kompleksitas P

$$P(x) = \left.\begin{array}{l} (\ldots(((a_n x + \\ a_{n-1})x + \\ a_{n-2})x + \\ a_{n-3})x + \\ \vdots \\ a_2)x + \\ a_1)x + \\ a_0 \end{array}\right\} \begin{array}{l} n \text{ penambahan} \\ n \text{ perkalian} \end{array}$$

$\rightarrow n + n$ operasi

$\therefore$ Kompleksitas kedua algoritma sama