

Nama	: Teosofi Hidayah Agung
NRP	: 5002221132

1. (Kompleksitas)

Nilai apa yang di-return oleh fungsi berikut? Nyatakan jawabanmu sebagai fungsi n.

Tentukan kompleksitas dari algoritma tersebut dalam notasi Big Oh.

```
function mystery(n)
r:=0
for i:= 1 to n - 1 do
    for j := i + 1 to n do
        for k := 1 to j do
            r := r + 1
        return(r)
```

Jawab:

Nilai yang di-return fungsi tersebut adalah perhitungan berapa banyaknya loop/perulangan pada fungsi tersebut. Misalkan $T(n)$ adalah fungsi yang mempresentasikan algoritma di atas.

$$\begin{aligned}
 T(n) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^j 1 = \sum_{i=1}^{n-1} \sum_{j=i+1}^n j = \sum_{i=1}^{n-1} \Delta^{-1} j^{(1)} \Big|_{i+1}^{n+1} = \sum_{i=1}^{n-1} \frac{j^{(2)}}{2} \Big|_{i+1}^{n+1} = \sum_{i=1}^{n-1} \frac{j(j-1)}{2} \Big|_{i+1}^{n+1} \\
 &= \sum_{i=1}^{n-1} \left(\frac{n(n+1)}{2} - \frac{i(i+1)}{2} \right) = \sum_{i=1}^{n-1} \frac{n(n+1)}{2} - \sum_{i=1}^{n-1} \frac{i(i+1)}{2} \\
 &= \frac{1}{2} [n(n+1)(n-1)] - \frac{1}{2} \left[\sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i \right] \\
 &= \frac{1}{2} [n^3 - n] - \frac{1}{2} \left[\frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2} \right] \\
 &= \frac{1}{2} \left[n^3 - n - \frac{2n^3 - 3n^2 + n}{6} - \frac{n^2 - n}{2} \right] \\
 &= \frac{1}{12} [6n^3 - 6n - 2n^3 + 3n^2 - n - 3n^2 + 3n] \\
 &= \frac{1}{12} [4n^3 - 4n] = \boxed{\frac{1}{3} [n^3 - n] \in O(n^3)}
 \end{aligned}$$

2. (Algoritma)

Konstruksi algoritma untuk mencari jarak terdekat antara dua elemen dalam suatu array bilangan.

```
MinDistance(A[0..n - 1])
//input: Array A[0.. n - 1] of numbers
//Output: Jarak minimum antara elemen-elemen di array
```

Jawab:

Algoritma yang akan dibuat adalah mengecek satu persatu selisih dari dua elemen kemudian memilih jarak terkecil antar salah satu pasangan elemen tersebut.

```
public static void MinDistance(int[] array){
    int min = Math.abs(array[0]-array[1]);
    for (int i = 0; i < array.length; i++) {
        for (int j = 0; j < i; j++) {
            int distance = Math.abs(array[j]-array[i]);
            if (distance<min) {
```

```

        min=distance;
    }
}
System.out.println("Jarak terdekat antara dua elemen adalah "+min);
} //jika lupa syntax nilai mutlak, dapat juga menggunakan definisi nilai
    mutlak

```

3. (Searching)

Buatlah coding dengan menggunakan algoritma binary search untuk mendapatkan nilai pembulatan dari akar bilangan bulat antara 0 dan 9.

petunjuk: gunakan method `int findsqrt(int x)` untuk menampilkan output:

```

sqrt(0) = 0
sqrt(1) = 1
sqrt(2) = 1
sqrt(3) = 1
sqrt(4) = 2
sqrt(5) = 2
sqrt(6) = 2
sqrt(7) = 2
sqrt(8) = 2
sqrt(9) = 3

```

Jawab:

Idenya adalah dengan menggunakan teori bilangan dimana ketika kita membagi bilangan x dengan 2 lalu dikuadratkan hasilnya, maka akan ada 3 kasus yaitu hasil kuadrat sama dengan, kurang dari, atau lebih dari x.

```

public static void main(String[] args) {
    for (int i = 0; i <= 9; i++) {
        System.out.println("sqrt("+i+") = "+findsqrt(i));
    }
}

public static int findsqrt(int x){
    int mid=0;
    int start=0;
    int end=x;
    int ans=0;
    while (start<=end) {
        mid = (start+end)/2;
        int square = mid*mid;
        if(square==x){
            return mid;
        }
        else if (square<x) {
            start=mid+1;
            ans=mid;
        }
        else{
            end=mid-1;
        }
    }
    return ans;
}

```

4. (Sorting)

Lengkapilah potongan program dibawah ini untuk mendapatkan tiga bilangan terbesar dari sebuah array [6, 4, 3, 4, 8, 2, 5] yaitu [5, 6, 8] berdasarkan algoritma *sorting* berikut:

(a) *Insertion sort*

```
1  public class FindThreeLargest{
2      public static void findThreeLargest(int[] arr)
3      {
4          int n = arr.length;
5          int largest1 = Integer.MIN_VALUE;
6          int largest2 = Integer.MIN_VALUE;
7          int largest3 = Integer.MIN_VALUE;
8
9          for (int i = 0; i < n; i++) {
10             int current = ...;
11
12             if(...){
13                 largest3 = largest2;
14                 largest2 = largest1;
15                 largest1 = current;
16             } else if (...&&...) {
17                 largest3 = largest2;
18                 largest2 = current;
19             } else if (...&&...) {
20                 largest3 = current;
21             }
22         }
23         System.out.println("Three largest elements;");
24         System.out.println(largest1);
25         System.out.println(largest2);
26         System.out.println(largest3);
27     }
28     public static void main(String[] args) {
29         int[] arr = {6,4,3,1,8,2,5};
30         findThreeLargest(arr);
31     }
32 }
```

Jawab:

```
10     int current = arr[i];
```

```
12     if(current>largest1)
```

```
16         else if (current<largest1 && current>largest2)
```

```
19             else if (current<largest2 && current>largest3)
```

(b) *Selection sort*

```
1  public class FindThreeLargest{
2
3      public static void findThreeLargest(int[] arr)
4      {
5          int n = arr.length;
```

```

6      int largest1 = Integer.MIN_VALUE;
7      int largest2 = Integer.MIN_VALUE;
8      int largest3 = Integer.MIN_VALUE;
9
10     for (int i = 0; i < n; i++) {
11         int largestIndex = i;
12         for (int j = i+1; j < n; j++) {
13             if (...) {
14                 largestIndex = ...;
15             }
16         }
17         if (...) {
18             largest3 = largest2;
19             largest2 = largest1;
20             largest1 = arr[largestIndex];
21         } else if (...&&...) {
22             largest3 = largest2;
23             largest2 = arr[largestIndex];
24         } else if (...&&...) {
25             largest3 = arr[largestIndex];
26         }
27
28         int temp = arr[i];
29         arr[i] = arr[largestIndex];
30         arr[largestIndex] = temp;
31     }
32     System.out.println("Three largest elements;");
33     System.out.println(largest1);
34     System.out.println(largest2);
35     System.out.println(largest3);
36 }
37 public static void main(String[] args) {
38     int[] arr = {6,4,3,1,8,2,5};
39     findThreeLargest(arr);
40 }
41 }

```

Jawab:

```
13     if (j>largestIndex)
```

```
14         largestIndex = j;
```

```
18     if (arr[largestIndex]>largest1)
```

```
22         else if (arr[largestIndex]<largest1 &&
                arr[largestIndex]>largest2)
```

```
25         else if (arr[largestIndex]<largest2 &&
                arr[largestIndex]>largest3)
```

5. (*Exception Handling*)

Perhatikan potongan program berikut:

```

1  String a;
2  String[] nama = new String[1];
3  try {
4      statment1;
5      nama[2] = "Linda";
6      statment2;
7  }
8  catch (ArithmeticException ex1) {
9      System.out.println(ex1);
10 }
11 catch (Exception ex2){
12     System.out.println(ex2);
13 }
14 finally {
15     System.out.println("bakso");
16 }
17 System.out.println("rawon");

```

Apa output dari program tersebut apabila terjadi kondisi seperti ini, jelaskan:

- a. Jika statment1 dan statment2 tidak terjadi exception (pernyataannya benar)

Jawab:

```

java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for
length 1
bakso
rawon

```

- b. Jika statment1 diganti dengan a=3; dan statment2 tidak terjadi exception

Jawab:

Program error saat di-running

- c. Jika statment1 tidak terjadi exception dan statment2 diganti dengan

System.out.println(1/0);

Jawab:

```

java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for
length 1
bakso
rawon

```

- d. Jika statment1 diganti dengan System.out.println(1/0); dan statment2 tidak terjadi exception

Jawab:

```

java.lang.ArithmeticException: / by zero
bakso
rawon

```