

Nama :
NPM :

SDA Genap 2015/2016

Analisis Algoritma

- Sebuah algoritma membutuhkan waktu 0.4ms untuk input sebesar 10. Berapa waktu yang dibutuhkan untuk input sebesar 100, apabila diketahui *running time*-nya adalah:
 - Linear
 - Quadratic
 - Cubic
 - Eksponential ($O(2^n)$)
- Sebuah algoritma dengan kompleksitas $O(\log N)$ membutuhkan 10 detik untuk memproses input berukuran $N = 32$. Perkiraan waktu yang dibutuhkan untuk memproses input berukuran $N = 128$?
- Untuk setiap *fragment code* berikut dan **input size n** , tentukanlah *Big-Oh* running time-nya!

a. Fragment #1

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

b. Fragment #2

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

c. Fragment #3

```
for(int i = 0; i < n; i++)  
    for(int j = 0; j < n; j++)  
        sum++;
```

d. Fragment #4

```
for(int i = 0; i < n; i++)  
    sum++;  
for(int j = 0; j < n; j++)  
    sum++;
```

e. Fragment #5

```
for(int i = 0; i < n; i++)  
    for(int j = 0; j < n*n; j++)  
        sum++;
```

f. Fragment #6

```
for(int i = 0; i < n; i++)  
    for(int j = i; j < n; j++)  
        sum++;
```

g. Fragment #7

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

h. Fragment #8

```
for(int i = 1; i < n; i++)  
    for (int j = 1; j < n; j=2*j)  
        sum++
```

- Berikan sebuah algoritma yang efisien untuk menentukan apakah terdapat suatu *integer* i sehingga $A[i] = i$ pada sebuah array of integers yang **terurut naik**. Berapakah *running time* dari algoritma Anda?
 - Terurut naik sejati
 - Terurut naik semu
- Diberikan sebuah array yang berisi elemen yang **monoton naik dan kemudian turun**. Contoh:
{1, 2, 3, 5, 9, 4, 3, 2}
{1, 11, 6, 4, 2}
{1, 4, 5, 10, 1}
Pada 3 array di atas, elemen 9, 11, dan 10 adalah "**elemen puncak**" pada array masing-masing. Nilai puncak hanya ada satu dan panjang array minimal adalah 3.
Berikan sebuah algoritma yang efisien untuk menentukan elemen puncak dari sebuah array yang mempunyai ciri-ciri tersebut !