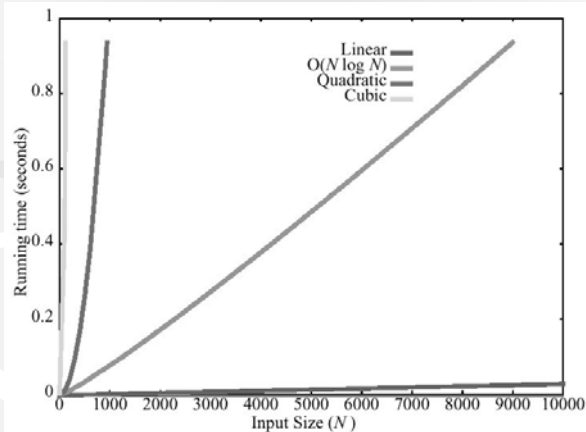


Function Curve



26-Feb-04

IKI10100 - PM

5 - 1

Contoh: $O(1)$ vs $O(N^2)$

```
FunctionA(int mat[][]) {  
    .....  
    .....  
    1000000 lines of code  
    .....  
    .....  
}  
O(1)
```

```
FunctionB(int mat[][]) {  
    x= # columns in mat;  
    y= # rows in mat;  
    .....  
    for(i=0;i<x;i++)  
        for(j=0;j<y;j++)  
            sum = mat[x][y];  
    .....  
    .....  
}  
O(n2)
```

26-Feb-04

IKI10100 - PM

5 - 2

MaxSubseq Problem

Dari sederet integer (bisa negatif), tentukan urutan bagiannya (*subsequent*) yang menghasilkan jumlah terbesar.

Contoh:

{-2, 11, -4, 13, -5, 2}

{1, -3, 4, -2, -1, 6}

26-Feb-04

IKI10100 - PM

5 - 3

MaxSubseq (1)

```
public static int maxSubsequenceSum(int[] a) {  
    int maxSum = 0;  
    for( int i=0; i < a.length; i++ )  
        for( int j=i; j < a.length; j++ ) {  
            int thisSum = 0;  
            for( int k=i; k <= j; k++ )  
                thisSum += a[k];  
            if( thisSum > maxSum ) {  
                maxSum = thisSum;  
                seqStart = i;  
                seqEnd = j;  
            }  
        }  
    return maxSum;  
}
```

26-Feb-04

IKI10100 - PM

5 - 4

MaxSubseq (2)

```
public static int maxSubsequenceSum( int[] a ) {
    int maxSum = 0;
    for( int i = 0; i < a.length; i++ ) {
        int thisSum = 0;
        for( int j = i; j < a.length; j++ ) {
            thisSum += a[j];
            if( thisSum > maxSum ) {
                maxSum = thisSum;
                seqStart = i;
                seqEnd = j;
            }
        }
    }
    return maxSum;
}
```

26-Feb-04

IKI10100 - PM

5 - 5

MaxSubseq (3)

```
public static int maxSubsequenceSum( int[] a ) {
    int maxSum = 0;
    int thisSum = 0;
    for( int i=0, j=0; j < a.length; j++ ) {
        thisSum += a[j];
        if( thisSum > maxSum ) {
            maxSum = thisSum;
            seqStart = i; seqEnd = j;
        }
        else if (thisSum < 0) {
            i = j + 1; thisSum = 0;
        }
    }
    return maxSum;
}
```

26-Feb-04

IKI10100 - PM

5 - 6

MaxSubseq (3)

Temukan urutan bilangan berikut yang menghasilkan jumlah terbesar dengan menerapkan algoritma linear.

3 5 1 0 -3 -2 -5 4 8 -6 7 1

26-Feb-04

IKI10100 - PM

5 - 7

Running Time of the 3 versions

N	$O(N^3)$	$O(N^2)$	$O(N \log N)$	$O(N)$
10	0.00103	0.00045	0.00066	0.00034
100	0.47015	0.01112	0.00486	0.00063
1,000	448.77	1.1233	0.05843	0.00333
10,000	NA	111.13	0.68631	0.03042
100,000	NA	NA	8.01130	0.29832

26-Feb-04

IKI10100 - PM

5 - 8

Latihan 1

- Sebuah algoritma perlu 30 detik untuk menyelesaikan permasalahan dengan 1000 input. Jika algoritma tersebut quadratic, berapa input size dapat diselesaikan dalam dua menit?

- a) 2000
- b) 4000
- c) 6000
- d) 60000
- e) salah semua

Latihan 2

- Sebuah algoritma perlu 6 detik untuk menyelesaikan permasalahan dengan input size 100 dan 10 menit untuk problem size 1000. Berapa running time dari algoritma tersebut?

- a) constant
- b) linear
- c) quadratic
- d) cubic
- e) salah semua

Various Growth Functions

- $T(N) = O(F(N))$ Growth of $T(N)$ is \leq growth of $F(N)$
- $T(N) = \Omega(F(N))$ Growth of $T(N)$ is \geq growth of $F(N)$
- $T(N) = \Theta(F(N))$ Growth of $T(N)$ is $=$ growth of $F(N)$
- $T(N) = o(F(N))$ Growth of $T(N)$ is $<$ growth of $F(N)$

Checkpoint 1

- Program A berjalan lebih cepat pada workstation X dibandingkan program B yang berjalan pada workstation Y.
- X dan Y punya arsitektur yang berbeda.
- A dan B adalah program Java yang berlainan fungsi.
- Apakah algoritma A lebih baik dari B ?

Checkpoint 2

- Program A berjalan lebih cepat pada workstation X dibandingkan program B yang berjalan pada workstation Y.
- X dan Y punya arsitektur yang berbeda.
- A dan B adalah program Java dengan fungsi yang sama.
- Apakah algoritma A lebih baik dari B ?

Checkpoint 3

- Dengan input size yang sama dan pada workstation yang sama, program A lebih cepat daripada program B yang $O(n^2)$.
- Dapat diambil kesimpulan: running time program A adalah linear atau logarithmic.
- Betul / Salah?

Tugas: Textbook

- Exercise 5.10, 5.11, 5.14, 5.15