



Xyba Project

----- Algoritma dan Pemrograman Pembahasan UAS 2016

1. This document is version: 0.9.9
Version should be at least 0.9 if you want to share this document to other people
2. You may not share this document if version is less than 1.0 unless you have my permission to do so
3. This document is created by Xyba, Student of Mathematics University of Indonesia Batch 2016
4. Should there be any mistakes or feedbacks you'd like to give, please contact me
5. Last Updated: 01/06/2018

Thank you for your cooperation >v<

1. Diberikan suatu vektor $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$. Buatlah algoritma untuk menentukan vektor satuan dari \mathbf{v} , yaitu:

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$$

dimana

$$\|\mathbf{v}\|_2 = \left(\sum_{i=1}^n |v_i|^2 \right)^{1/2}$$

Jawab:

```
input array  $\mathbf{v}$  with length  $n$ 
Norm2 =  $v_1^2$ 
for  $i = 2$  to  $n$ :
    Norm2 = Norm2 +  $v_i^2$ 
endfor
Norm2 = Sqrt[Norm2]
for  $i = 1$  to  $n$ :
     $\mathbf{v}[i] = \frac{\mathbf{v}[i]}{\text{Norm2}}$ 
endfor
return  $\mathbf{v}$ 
```

2. Buatlah suatu algoritma untuk membalikkan (*reverse*) urutan isi suatu *array* **a** dengan banyak elemen n dan tentukan *big-Oh* dari algoritma yang anda buat. Berikut adalah ilustrasi perubahan elemen dari **a**:

a_1	a_2	a_3	...	a_{n-1}	a_n
-------	-------	-------	-----	-----------	-------

 menjadi

a_n	a_{n-1}	...	a_3	a_2	a_1
-------	-----------	-----	-------	-------	-------

Jawab:

- Berikut algoritma untuk menentukan algoritma untuk membalikkan urutan isi suatu array **a**.

```

input a = [ $a_1, a_2, \dots, a_n$ ]
for  $i = 1$  to  $\lfloor \frac{n}{2} \rfloor$ :
     $x = n - i + 1$ 
     $c = \mathbf{a}[i]$ 
     $\mathbf{a}[i] = \mathbf{a}[x]$ 
     $\mathbf{a}[x] = c$ 
endfor
return a

```

- Akan ditentukan *big-Oh* dari algoritma tersebut

Pertama, akan ditentukan kompleksitas waktu dari algoritma tersebut

	Waktu	Jumlah Operasi
input a = [a_1, a_2, \dots, a_n]		
for $i = 1$ to $\lfloor \frac{n}{2} \rfloor$:	t_1	$\lfloor \frac{n}{2} \rfloor$
$x = n - i + 1$	t_2	$3 \lfloor \frac{n}{2} \rfloor$
$c = \mathbf{a}[i]$	t_3	$2 \lfloor \frac{n}{2} \rfloor$
$\mathbf{a}[i] = \mathbf{a}[x]$	t_4	$2 \lfloor \frac{n}{2} \rfloor$
$\mathbf{a}[x] = c$	t_5	$\lfloor \frac{n}{2} \rfloor$
endfor		
return a	t_6	1

Sehingga kompleksitas waktu dari algoritma tersebut diberikan oleh:

$$\begin{aligned}
 T(n) &= t_1 \lfloor \frac{n}{2} \rfloor + 3t_2 \lfloor \frac{n}{2} \rfloor + 2t_3 \lfloor \frac{n}{2} \rfloor + 2t_4 \lfloor \frac{n}{2} \rfloor + t_5 \lfloor \frac{n}{2} \rfloor + t_6 \\
 &= (t_1 + 3t_2 + 2t_3 + 2t_4 + t_5) \lfloor \frac{n}{2} \rfloor + t_6
 \end{aligned}$$

Perhatikan bahwa:

$$\begin{aligned} T(n) &= (t_1 + 3t_2 + 2t_3 + 2t_4 + t_5) \left\lfloor \frac{n}{2} \right\rfloor + t_6 \leq (t_1 + 3t_2 + 2t_3 + 2t_4 + t_5) \left(\frac{n}{2} \right) + t_6 \\ &= \left(\frac{t_1}{2} + \frac{3t_2}{2} + t_3 + t_4 + \frac{t_5}{2} \right) n + t_6 \\ &\leq \left(\frac{t_1}{2} + \frac{3t_2}{2} + t_3 + t_4 + \frac{t_5}{2} \right) n + t_6 \cdot n \\ &= \left(\frac{t_1}{2} + \frac{3t_2}{2} + t_3 + t_4 + \frac{t_5}{2} + t_6 \right) n, \forall n \geq 1 \end{aligned}$$

Sehingga, ambil $c = \frac{t_1}{2} + \frac{3t_2}{2} + t_3 + t_4 + \frac{t_5}{2} + t_6$ dan $n_0 = 1$, maka kita akan peroleh:
$$0 \leq T(n) \leq c \cdot n, \forall n \geq n_0$$

Sehingga berdasarkan definisi, maka $T(n) \in O(n)$.

Artinya algoritma tersebut memiliki *big-Oh* $O(n)$.

3. Diberikan suatu fungsi $f(x) = 2x^{10} - x^3 + 2x - 20$. Buatlah algoritma untuk menentukan nilai maksimum fungsi tersebut pada nilai-nilai $[-n, -n + 0.5, \dots, -1, -0.5, 0, 0.5, 1, \dots, n - 0.5, n]$.

Petunjuk: buatlah fungsi $f(x)$ dengan perintah FUNCTION. Selanjutnya, cari nilai maksimum fungsi tersebut pada nilai-nilai di atas.

Jawab:

```
function f(x)
    return  $2x^{10} - x^3 + 2x - 20$ 
endfunction
```

```
input n
Max = f(n)
for i = 1 to 4n:
    n = n - 0.5
    t = f(n)
    if t > Max:
        Max = t
    endif
endfor
return Max
```

4. Sebuah barisan DNA terdiri dari huruf A, T, G, dan C. Buatlah algoritma untuk menentukan jumlah A, T, G, dan C dalam suatu barisan DNA dengan panjang barisan n . Berikut adalah contoh sebuah barisan DNA:

TACCGCGGCTATTACTGCCAGGAAGGAACT

Jawab:

```
input DNA string " $d_1d_2 \dots d_n$ "
Count = [0,0,0,0]
for  $i = 1$  to  $n$ 
    if  $d_i = "A"$ :
        Count[1] = Count[1] + 1
    else if  $d_i = "T"$ :
        Count[2] = Count[2] + 1
    else if  $d_i = "G"$ :
        Count[3] = Count[3] + 1
    else:
        Count[4] = Count[4] + 1
    endif
endfor
print("banyaknya huruf A pada DNA tersebut adalah ", Count[1])
print("banyaknya huruf T pada DNA tersebut adalah ", Count[2])
print("banyaknya huruf G pada DNA tersebut adalah ", Count[3])
print("banyaknya huruf C pada DNA tersebut adalah ", Count[4])
```

5. Diberikan suatu matriks A berukuran $m \times n$. Buatlah algoritma untuk menentukan *norm Frobenius* dari A yang didefinisikan sebagai berikut:

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2}$$

dimana a_{ij} adalah elemen ij dari matriks A .

Jawab:

```
input A = [[a11, a12, ..., a1n],  
           [a21, a22, ..., a2n],  
           ...  
           [am1, am2, ..., amn]]  
Frob = 0  
for i = 1 to m:  
    for j = 1 to n:  
        Frob = Frob + aij2  
return Sqrt[Frob]
```

6. Buatlah algoritma untuk mencari elemen terbesar dari suatu matriks A berukuran $n \times n$ beserta posisinya (i, j) dari elemen terbesar tersebut.

Catatan: elemen terbesar dapat muncul lebih dari 1 kali

Jawab:

Asumsikan bahwa soal meminta untuk mengoutput semua posisi elemen-elemen terbesar berdasarkan catatan yang diberikan jika ada lebih dari satu elemen terbesar dari matriks A .

```
input A = [[a11, a12, ..., a1n],
           [a21, a22, ..., a2n],
           ...
           [an1, an2, ..., ann]]
Max = a11
Pos = [ ]
for i = 1 to n:
    for j = 1 to n:
        el = aij
        if el > Max:
            Max = el
            Pos = [(i, j)]
        elif el = Max:
            k = Length[Pos]
            Pos[k + 1] = (i, j)
        endif
    endfor
endfor
print("elemen maksimum dari A adalah ", Max)
if Length[Pos] = 1:
    print("posisi dari elemen maksimum tersebut adalah: ", Pos[1])
else:
    print("posisi-posisi dari elemen maksimum tersebut yaitu: ", Pos)
```


Afterword

Pembuatan dokumen ini dibantu oleh:

1. Vin G, Matematika UI 2017.