**SOAL DAN JAWABAN UTS**

**MATA KULIAH ANALISIS ALGORITMA**



Disusun oleh:

Nama : Muhammad Fadillah Arsa

NPM : 140810170005

Kelas : A

Program Studi : S1 Teknik Informatika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2019**

**SOAL DAN JAWABAN UTS**

**MATA KULIAH ANALISIS ALGORITMA**

**NOMOR 1**

**1.A. Carilah masig-masing T(n), kompleksitas O, c, dan n0!**

for j = 1 to n-1 -> looping j, sebanyak n-1

k = j -> 1 operasi assignment

for i = j + 1 to n -> looping i

if a[i ] < a[k] then -> 1 operasi perbandingan, worst case: selalu true

k = i -> 1 operasi assignment

endif

endfor

tm = a[j] -> 1 operasi assignment

a[j] = a[k] -> 1 operasi assignment

a[k] = tm -> 1 operasi assignment

endfor

|  |  |
| --- | --- |
| **J** | **Jumlah Looping i** |
| 1 | n-1 |
| 2 | n-2 |
| … | … |
| n-2 | 2 |
| n-1 | 1 |

* Mencari T(n)

T(n) = (operasi di dalam blok i \* jumlah looping i terkait looping j) + (operasi di luar blok i \* jumlah looping j)

T(n) = ( 2 \* ((a+Un) ) + 4 (n-1)

T(n) = (2 \* ((1+n-1) + 4n – 4 )

T(n) = n2 - n + 4n – 4

**T(n) = n2 + 3n - 4**

* Mencari Kompleksitas O

**Kompleksitas O = O (n2)**

* Mencari c dan n0

T(n) ≤ c . f(n)

n2 + 3n – 4 ≤ c . n2

1 + - ≤ c

Maka:

**n0 = 2** (memenuhi n0 ≥ 0)

**c= 1,5** (memenuhi c > 0)

**1.B. Carilah masig-masing T(n), kompleksitas O, c, dan n0!**

for i = 0 to n-1 -> looping i, sejumlah n

for j = 0 to n-1 -> looping j, sejumlah n

c[i,j] = 0 -> 1 operasi assignment

for k = 0 to n-1 -> looping k, sejumlah n

cij = d [i,k] and b[k,j] -> 1 operasi assignment

c[i,j] = c[i,j] or cij -> 1 operasi assignment

endfor

endfor

endfor

* Mencari T(n)

T(n) = ((( 2 \* n ) + 1 ) \* n ) n

T(n) = (( 2n+1 ) n) n

T(n) = ( 2n2 + n ) n

**T(n) = 2n3 + n2**

* Mencari Kompleksitas O

**Kompleksitas O = O (n3)**

* Mencari c dan n0

T(n) ≤ c . f(n)

2n3 + n2 ≤ c . n3

2 + ≤ c

Maka:

**n0 = 1** (memenuhi n0 ≥ 0)

**c= 3** (memenuhi c > 0)

**NOMOR 2**

**2.A. Cari masing-masing T(n), kompleksitas O, c, dan no!**

**Algoritma (a)**

ada = 0 -> 1 operasi assignment

kx = 1 -> 1 operasi assignment

input br; -> 1 operasi assignment (berupa input yang dipakai dalam program)

for (i = 1; i < n+1; i++) { -> looping i, maksimal sebanyak n

if (a[i] == br && (!ada)) { -> 3 operasi perbandingan

ada = 1; -> 1 operasi assignment

kx = I; -> 1 operasi assignment

i = n+1; -> 1 operasi penjumlahan, dan 1 operasi assignment

}

}

Worst Case:

Proses perbandingan if bernilai false sebanyak n-1, pada iterasi ke n bernilai true dan membuat i !< n+1.

* Mencari T(n)

T(n) = 3 + ( 3 \* (n-1) + (3+4) )

T(n) = 3 + ( 3n – 3 + 4 )

**T(n) = 3n + 7**

* Mencari Kompleksitas O

**Kompleksitas O = O (n)**

* Mencari c dan n0

T(n) ≤ c . f(n)

3n + 7 ≤ c . n

3 + ≤ c

Maka:

**n0 = 1** (memenuhi n0 ≥ 0)

**c= 10** (memenuhi c > 0)

**Algoritma (b)**

L = 1 -> 1 operasi assignment

R = n -> 1 oparasi assignment

ada = 0 -> 1 operasi assignment

input br; -> 1 operasi assignment (berupa input yang dipakai dalam program)

while ((L<=R) && (!ada)) { -> 3 operasi perbandingan, while sebanyak log2 n

m = (L+R) div 2; -> 2 operasi perhitungan, 1 operasi assignment

if (a[m] == br) -> 1 operasi perbandingan

ada = 1; -> 1 operasi assignment

else if (br < a[m]) -> 1 operasi perbandingan

R = m-1 -> 1 operasi assignment

else

L = m+1; -> 1 operasi penjumlahan dan 1 operasi assignment

}

Worst Case Blok If: operasi selalu bernilai true pada blok else (3 operasi).

* Mencari T(n)

T(n) = 4 + ( 3+3+4 \* log2 n)

T(n) = 10 log2 n + 4

* Mencari Kompleksitas O

**Kompleksitas O = O (log2 n)**

* Mencari c dan n0

T(n) ≤ c . f(n)

10 log2 n + 4 ≤ c . log2 n

10 + ≤ c

Maka:

**n0 = 2** (memenuhi n0 ≥ 0)

**c= 14** (memenuhi c > 0)

**2.B. Komputer A mengeksekusi 109 instruksi per detik, Komputer B 107 instruksi per detik. Komputer A akan menggunakan algoritma 2.a. dan komputer B menggunakan algoritma 2.b.. Data yang digunakan sebanyak 108. Hitung running time masing-masing menggunakan algoritma kompleksitas O, dan algoritma mana yang labih baik?**

**Jumlah Instruksi Yang Diproses Algoritma A**

I = n (kompleksitas algoritma A)

I = 108

**Jumlah Instruksi Yang Diproses Algoritma B**

I = log2 n (kompleksitas algoritma B)

I = log2 108 = 26,5

**Running Time Komputer A**

RTA = 108 / 109 = 0,1 second

**Running Time Komputer B**

RTB = 26,5 / 107 = 26,5 x 10-7 second

RTA>RTB

**Jadi, algoritma yang lebih baik adalah algoritma B.**.