**UJIAN TENGAH SEMESTER**

**PRAKTIKUM ANALISIS ALGORITMA**

****

**Muhammad Luthfiansyah**

**140810170023**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS PADJADJARAN**

**JATINANGOR**

**2019**

1.

// Soal 1.a

for j = 1 to n-1

k = j n-1 kali

for i = j + 1 to n

if a[i] < a[k] then ∑\_(i=1)^(n-1)▒i

k = I ∑\_(i=1)^(n-1)▒i

endif

endfor

tm = a[j] n-1 kali

a[j] = a[k] n-1 kali

a[k] = tm n-1 kali

endfor

T(n)=(n-1)+(n-1)+∑\_(i=1)^(n-1)▒i+∑\_(i=1)^(n-1)▒i+(n-1)+(n-1)+(n-1)

=4(n-1)+2∑\_(i=1)^(n-1)▒i

=4n-4+2(1/2 (n-1)(n))

=4n-4+n^2-n

=n^2+3n-4

=O(n^2)

T(n)≤c .f(n)

n^2+3n-4≤c .n^2

1+3/n-4/n^2 ≤c

mis n\_0=1

1+3/1-4/1≤c

c≥0

Maka, c≥0, sehingga syarat n\_0 adalah positif dan c adalah positif terpenuhi.

// Soal 1.b

for i = 0 to n-1

for j = 0 to n-1

c[i,j] = 0 n^2 kali

for k = 0 to n-1

cij = d[i,k] and b[k,j] n^3 kali

c[i,j] = c[i,j] or cij n^3 kali

endfor

endfor

endfor

T(n)=n^2+n^3+n^3

=2n^3+n^2

=O(n^3)

T(n)≤c .f(n)

2n^3+n^2≤c .n^3

2+1/n≤c

mis n\_0=1

2+1/1≤c

c≥3

Maka, c≥3, sehingga syarat n\_0 adalah positif dan c adalah positif terpenuhi.

2.

// Soal 2.a

Algoritma (a)

ada = 0 //1

kx = 0 //1

input br

for(i=1;i<n+1;i++){

if(a[i]==br && (!ada)){ // Penjelasan A

ada = 1 //1

kx = I //1

i = n+1 //1

}

}

Misal n = 5

Penjelasan A

Worst casenya adalah ketika setiap perbandingan nilainya selalu salah sampai akhirnya yang ditemukan ada pada posisi terujung. Berikut adalah ilustrasinya

I Kondisi

1 NOT FOUND

2 NOT FOUND

3 NOT FOUND

4 NOT FOUND

5 FOUND

Maka akan terjadi proses perbandingan sebanyak n

T(n)=n+5

T(n)≤c\*f(n)

n+5≤c\*n

1+5/n≤c (Misal n0=1)

1+5≤c

c≥6

Maka, c≥6, sehingga syarat n\_0 adalah positif dan c adalah positif terpenuhi.

Algoritma (b)

L = 1 //1

R = n //1

ada = 0 //1

input br

while((L<=R)&&(!ada)){ //Seluruh While masuk kedalam Penjelasan A

m = (L+R) div 2 // log n

if(a[m]==br) // log n

ada = 1 // 1

else if(br<a[m]) // baris ini sampai akhir jumlahnya 2(log n -1)

R = m-1

else

L = m+1

}

Penjelasan A

Bagian tersebut merupakan bagian dari binary searching

k(banyak langkah) Jumlah proses

1 n/2

2 n/2^2

3 n/2^3

4 n/2^4

5 n/2^5

Apabila dalam worst case, maka akan dibagi sampai menyisakan 1. Sehingga dapat disimpulkan bahwa

n/2^k =1

n=2^k

log\_2⁡n=k

T(n)1=log\_2⁡n

T(n)= 4(log\_2⁡〖n)〗+2

T(n)≤c\*f(n)

4(log\_2⁡〖n)〗+2≤c\* log\_2⁡n

4+2/log\_2⁡n ≤c (Misal n0=2)

4+2≤c

c≥6

Maka, c≥6, sehingga syarat n\_0 adalah positif dan c adalah positif terpenuhi.

// Soal 2.b

Komputer A

Kompleksitas O(n)

Kecepatan Komputer 109 instruksi / detik

Data sebanyak 108 data

Running time= (Jumlah instruksi)/(Kecepatan Komputer)

Running time= 〖10〗^8/〖10〗^9 =0,1 detik

Komputer B

Kompleksitas O(log2(n))

Kecepatan Komputer 107 instruksi / detik

Data sebanyak 108 data

Running time= (Jumlah instruksi)/(Kecepatan Komputer)

Running time= log\_2⁡〖〖10〗^8 〗/〖10〗^7 =26,5754/〖10〗^7

Running time=2,65754\*〖10〗^(-6) detik

Maka algoritma terbaik adalah algoritma (b), karena meski kecepatan computer yang lebih lambat dari computer A, running timenya lebih kecil.