**TUGAS PRAKTIKUM**

**UTS**

**MATA KULIAH ANALISIS ALGORITMA D10G.4205 & D10K.0400601**



Muhammad Ariq Farhansyah Mutyara

140810170053

**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

k = I

endif

endfor

tm = a[j] n-1 kali

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

a[k] = tm n-1 kali

endfor

Maka, , sehingga syarat adalah positif dan adalah positif terpenuhi.

**// Soal 1.b**

for i = 0 to n-1

for j = 0 to n-1

c[i,j] = 0 kali

for k = 0 to n-1

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

c[i,j] = c[i,j] or cij kali

endfor

endfor

endfor

Maka, , sehingga syarat adalah positif dan 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

Maka, , sehingga syarat adalah positif dan 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 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

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

Maka, , sehingga syarat adalah positif dan adalah positif terpenuhi.

**// Soal 2.b**

* Komputer A

Kompleksitas O(n)

Kecepatan Komputer 109 instruksi / detik

Data sebanyak 108 data

* Komputer B

Kompleksitas O(log2(n))

Kecepatan Komputer 107 instruksi / detik

Data sebanyak 108 data

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