LKP 2 Analgor

Halida Fiadnin - G6401211142

No

Date

1) dit: tentukan dominant terms dan kompleksitas Big-Oh dari setiap algoritma

| expression | dominant term(s) | O(...) |
|---|---|---|
| $5 + 0.001n^3 + 0.025n$ | $0.001\,n^3$ | $O(n^3)$ |
| $500n + 100n^{1.5} + 50n\log_{10} n$ | $100n^{1.5}$ | $O(n^{1.5})$ |
| $100n + 0{,}01n^2$ | $0.01n^2$ | $O(n^2)$ |
| $0.01n + 100n^2$ | $100n^2$ | $O(n^2)$ |
| $2n + n^{0.5} + 0.5n^{1.25}$ | $0.5n^{1.25}$ | $O(n^{1.25})$ |
| $100n\log_3 n + n^3 + 100n$ | $n^3$ | $O(n^3)$ |
| $0.003\log_4 n + \log_2\log_2 n$ | $0.03\log_4 n$ | $O(\log n)$ |

2) a. dik: $T_A(n) = 0.1n\log_2 n$ mikrodetik    dit: mana yang lebih baik?

$T_B(n) = 5n$ mikrodetik

$n = 10^{12}$

jawab: $T_A(n) = 0.1\,n\log_2 n = 0.1 \times 10^{12} \times \log_2 10^{12}$

$= 10^{11} \times 39{,}863$

$= 3{,}986 \times 10^{12}$ mikrodetik

$T_B(n) = 5 \times 10^{12}$ mikrodetik

$\Rightarrow 3{,}986 \times 10^{12} < 5 \times 10^{12}$

$T_A(n) < T_B(n)$

$\therefore$ $T_A(n)$ lebih baik dari $T_B(n)$ karena waktu prosesnya lebih cepat //

b. dik: $T_A(n) = 0{,}001\,n$ milidetik    dit: mana yang lebih baik?

$T_B(n) = 500\sqrt{n}$ milidetik

$n = 10^9$

jawab: $T_A(n) = 10^{-3} \times 10^9$

$= 10^6$ milidetik

$T_B(n) = 500 \times \sqrt{10^9}$

$= 5 \times 10^2 \times 10^4 \times \sqrt{10}$

$= 5\sqrt{10} \times 10^6$ milidetik

$\Rightarrow 10^6 < 5\sqrt{10} \times 10^6$

$T_A(n) < T_B(n)$

$\therefore$ $T_A(n)$ lebih baik dari $T_B(n)$ karena waktu prosesnya lebih cepat //

3) Tentukan F(n) dan kompleksitas

a)
```
for(i=0; i<n; i++) {
    for(j=0; j<n; j++) {
        for (k=0; k<n; k++) {
            ... //operasi konstan
        }
    }
}
```

No
Date

| loop | cost | operation | times |
|---|---|---|---|
| | $c_1$ | i=0 | 1 |
| 1 | $c_2$ | i<n | n+1 |
| | $c_3$ | i++ | n |
| | $c_4$ | j=0 | $n*1$ |
| 2 | $c_5$ | j<n | $n*(n+1)$ |
| | 66 | j++ | $n*n$ |
| | $c_7$ | k=0 | $n*n*1$ |
| 3 | $c_8$ | k<n | $n*n*(n+1)$ |
| | $c_9$ | k++ | $n*n*n$ |
| | $c_{10}$ | operasi konstan | $n*n*n*c$ |

$\Sigma$ operasi loop 1 = $1+(n+1)+n$
$= 2n+2$

$\Sigma$ operasi loop 2 = $n+(n^2+n)+(n^2)$
$= 2n^2+2n$

$\Sigma$ operasi loop 3 = $n^2+(n^3+n^2)+(n^3)$
$= 2n^3+2n^2$

operasi konstan = $n^3c$

$F(n) = 2n+2+2n^2+2n+2n^3+2n^2+n^3c$
$= 2n^3+4n^2+4n+2+n^3c$
$\boxed{F(n) = (2+c)n^3+4n^2+4n+2}$

$\boxed{\text{complexity} = O(n^3)}$

b)
```
for(i=n; i>0; i/=2) {
    for(j=1; j<n; j*=2 {
        for(k=0; k<n; k+=2{
            ... // operasi konstan
        }
    }
}
```

| loop | cost | operation | times |
|---|---|---|---|
| | $c_1$ | i=n | 1 |
| 1 | $c_2$ | i>0 | $\log_2(n)+2$ |
| | $c_3$ | i/=2 | $\log_2(n)+1$ |
| | $c_4$ | j=1 | $(\log_2(n)+1)\times 1$ |
| 2 | $c_5$ | j<n | $(\log_2(n)+1)\times(\log_2(n)+1)$ |
| | 66 | j*=2 | $(\log_2(n)+1)\times(\log_2(n))$ |
| | $c_7$ | k=0 | $(\log_2(n)+1)\times(\log_2(n))\times 1$ |
| 3 | $c_8$ | k<n | $(\log_2(n)+1)\times(\log_2(n))\times\left(\frac{n+1}{2}+1\right)$ |
| | $c_9$ | k+=2 | $(\log_2(n)+1)\times(\log_2(n))\times\left(\frac{n+1}{2}\right)$ |
| | $c_{10}$ | constanta | $(\log_2(n)+1)\times(\log_2(n))\times\left(\frac{n+1}{2}\right)\times c$ |

No
Date

$$\Sigma \text{ operasi loop 1} = 1 + \log_2(n) + 2 + \log_2(n) + 1$$
$$= 2\log_2(n) + 4$$

$$\Sigma \text{ operasi loop 2} = (\log_2(n)+1) + (\log_2(n)^2 + 2\log_2(n)+1) + (\log_2(n)^2 + \log_2(n))$$
$$= 2\log_2(n)^2 + 4\log_2(n) + 2$$
$$= 2(\log_2(n)+1)^2$$

$$\Sigma \text{ operasi loop 3} : (\log_2(n)^2 + 2\log_2(n)+1) + \left(\left(\log_2(n)^2 + \log_2(n)\right) \times \left(\frac{n+3}{2}\right)\right) + \left(\left(\log_2(n)^2 + \log_2(n)\right) \times \left(\frac{n+1}{2}\right)\right)$$
$$= \left(\frac{2\log_2(n)^2 + 4\log_2(n) + 2}{2}\right) + \left(\frac{(n+3)\log_2(n)^2 + (n+3)\log_2(n)}{2}\right) + \left(\frac{(n+1)\log_2(n)^2 + (n+1)\log_2(n)}{2}\right)$$

$$= \frac{(2n+6)\log_2(n)^2 + (2n+8)\log_2(n) + 2}{2}$$

$$\Sigma \text{ constanta} = c \cdot \left(\frac{(2n+6)\log_2(n)^2 + (2n+8)\log_2(n)+2}{2}\right)$$

$$F(n) = \frac{4\log_2(n)+16}{2} + \frac{4\log_2(n)^2 + 8\log_2(n)+4}{2} + \frac{(2n+6)\log_2(n)^2 + (2n+8)\log_2(n)+2}{2} + \frac{(2n+6)\log_2(n)^2 + (2n+8)\log_2(n)+2}{2} \times c$$

$$= \frac{(2n+10+2nc+6c)\log_2(n)^2 + (2n+20+2nc+8c)\log_2(n) + 22}{2}$$

$$F(n) = (n+5+nc+3c)\log_2(n)^2 + (n+10+nc+4c)\log_2(n) + 11$$

$$\text{complexity} = O(\log_2(n)^2 \cdot n)$$
$$= O(n \cdot \log_2(n) \cdot \log_2(n))$$

c)
```
For(bound=1; bound<=n; bound *= 2) {
    For(j=0; j<n; j+=2) {
        ... // operasi konstan
    }
    For(j=1; j<n; j*=2) {
        ... // operasi konstan
    }
}
```

No ..............................
Date

| loop | cost | operation | times |
|---|---|---|---|
| | $c_1$ | bound = 1 | $1$ |
| 1 | $c_2$ | bound ≤ n | $\log_2(n)+2$ |
| | $c_3$ | bound *= 2 | $\log_2(n)+1$ |
| | $c_4$ | j = 0 | $(\log_2(n)+1) \times 1$ |
| 2 | $c_5$ | j < n | $(\log_2(n)+1) \times \left(\frac{n+1}{2}+1\right)$ |
| | $c_6$ | j += 2 | $(\log_2(n)+1) \times \left(\frac{n+1}{2}\right)$ |
| | $c_7$ | c | $(\log_2(n)+1) \times \left(\frac{n+1}{2}\right) \times c$ |
| | $c_8$ | j = 1 | $(\log_2(n)+1) \times 1$ |
| 3 | $c_9$ | j < n | $(\log_2(n)+1) \times (\log_2(n)+2)$ |
| | $c_{10}$ | j *= 2 | $(\log_2(n)+1) \times (\log_2(n)+1)$ |
| | $c_{11}$ | c | $(\log_2(n)+1) \times (\log_2(n)+1) \times c$ |

$$\Sigma \text{operasi loop 1} = 1 + (\log_2(n)+2) + (\log_2(n)+1)$$
$$= 2\log_2 n + 4$$

$$\Sigma \text{operasi loop 2} = (\log_2(n)+1) + \left(\log_2(n) \times \left(\frac{n+3}{2}\right) + \left(\frac{n+3}{2}\right)\right) + \left(\log_2(n) \times \left(\frac{n+1}{2}\right) + \left(\frac{n+1}{2}\right)\right)$$
$$+ \left(\log_2(n) \times \left(\frac{n+1}{2}\right) \times c + \left(\frac{n+1}{2}\right) \times c\right)$$
$$= \frac{(2n+6+nc+c)\log_2(n) + (2n+6+nc+c)}{2}$$

$$\Sigma \text{operasi loop 3} = (\log_2(n)+1) + (\log_2(n)^2 + 3\log_2(n) + 2) + (\log_2(n)^2 + 2\log_2(n)+1)$$
$$+ (\log_2(n)^2 + 2\log_2(n) + 1) \times c$$
$$= (2+c)\log_2(n)^2 + (6+2c)\log_2(n) + (4+c)$$

$$F(n) = \frac{4\log_2 n + 8}{2} + \frac{(2n+6+nc+c)\log_2(n) + (2n+6+nc+c)}{2} + \frac{(4+c)\log(n)^2 + (12+4c)\log_2(n) + (8+2c)}{2}$$

$$F(n) = \frac{(4+c)\log_2(n)^2 + (2n+22+nc+5c)\log_2(n) + (2n+22+nc+3c)}{2}$$

Complexity = $O(\log_2(n)^2)$

$O(\log_2(n) \cdot \log_2(n))$

Lalu hitunglah F(n) dari masing-masing potongan berikut ini, manakah algoritme yang lebih baik? Algoritme rata-1 atau rata-2? (hint: kali = diisi dengan berapa kali instruksi dijalankan, waktu = diisi dengan berapa satuan waktu yang dibutuhkan)

(a) Perbandingan pertama

| Kelas-kelas instruksi | Asumsi waktu |
|---|---|
| 1. instruksi matematis | 1 satuan waktu |
| 2. instruksi logika, assign | ½ satuan waktu |
| 3. instruksi I/O dan increment otomatis | 2 satuan waktu |

**Semakin detil dan teliti pengelompokan dan pendefinisian asumsi waktu → semakin teliti f(n) yang dihasilkan**

| Instruksi | Kali | waktu | Hasil | |
|---|---|---|---|---|
| Algoritma Rata-1 : real | | | | |
| Begin | | | | |
| Jml <= 0 | 1 | ½ sw | ½ | |
| For i <= 1 to n | n | 2 sw | 2n | F(n)=3½n+2 |
| Do jml <= jml + A[ i ] | n | 1½ sw | 1½n | |
| End-for | | | | |
| Rata_1 <= jml/n | 1 | 1½ sw | 1½ | |
| End-Alg1 | | | | |
| Algoritma Rata-2 : real | | | | |
| Begin | | | | |
| i <= 1 | 1 | ½ sw | ½ | |
| Jml <= 0 | 1 | ½ sw | ½ | |
| While i ≤ n | n+1 | ½ sw | ½n+½ | |
| Do jml <= jml + A[ i ] | n | 1½ sw | 1½n | F(n)=3½n+3 |
| i <= i + 1 | n | 1½ sw | 1½n | |
| End-while | | | | |
| Rata_2 <= jml/n | 1 | 1½ sw | 1½ | |
| End-Alg2 | | | | |

(b) Perbandingan kedua

| Instruksi | kali | waktu | Hasil | |
|---|---|---|---|---|
| Algoritma Rata-1 : real | | | | |
| Begin | | | | |
| Jml <= 0 | 1 | ½ sw | ½ | |
| For i <= 1 to n | n | 2 sw | 2n | |
| Do jml <= jml + A[ i ] | n | 1½ sr | 1½n | F(n)=3½n+2 |
| End-for | | | | |
| Rata_1 <= jml/n | 1 | 1½ s | 1½ | |
| End-Alg1 | | | | |
| Algoritma Rata-2 : real | | | | |
| Begin | | | | |
| i <= 1 | 1 | ½ sw | ½ | |
| Jml <= 0 | 1 | ½ sw | ½ | |
| While i ≤ n | n+1 | ½ sw | ½n+½ | |
| Do jml <= jml + A[ i ] | n | 1½ sw | 1½n | F(n)=3½n+3 |
| i <= i + 1 | n | 1½ sw | 1½n | |
| End-while | | | | |
| Rata_2 <= jml/n | 1 | 1½ sw | 1½ | |
| End-Alg2 | | | | |

Selamat Mengerjakan