

Lembar Kerja Responsi 2
Mata Kuliah KOM 401 Analisis Algoritme
Semester Ganjil Tahun Akademik 2020/2021

Nama : Shibgotalloh Sabilana
NIM : G64180002

1. $f(n)$ sangat membantu kita memahami seberapa baik suatu algoritme. Namun, pada prakteknya $f(n)$ dapat disederhanakan. Oleh karena itu, sederhanakan $f(n)$ berikut:

- A. n^4
- B. n^8
- C. n^3
- D. n^3

2. Urutkan kompleksitas persamaan di bawah ini dari yang terkecil

- a. 102
- b. $n!$
- c. $n^3 + 6$
- d. $\log(n)$
- e. $\sqrt{n} - 4$
- f. n^n
- g. $n + 4$
- h. 2^n

Jawaban :

a, d, e, g, c, h, b, f

3. Menggunakan pembuktian kontradiksi, tunjukkan bahwa setidaknya terdapat 5 hari yang sama dari 29 hari

Dalam satu minggu terdapat 7 hari, maka jika dalam 29 hari ingin menemukan terdapat 5 hari yang sama bisa ditemukan dengan cara pengurangan per minggu

Minggu 1 = 7

Minggu 2 = 7

Minggu 3 = 7

Minggu 4 = 7

Minggu 5 = 1

Maka dipastikan setidaknya terdapat satu hari yang memiliki 5 hari yang sama dalam 29 hari

4. Buktikan dengan induksi matematika bahwa:

$$P I \quad n = 1 \quad n^3 = \left(\frac{n}{2} (n+1) \right)^2$$

$$1^3 = \left(\frac{1}{2} (2) \right)^2$$

$$1 = 1$$

terbukti

$$P II \quad n = k \quad k^3 = \left(\frac{k}{2} (k+1) \right)^2$$

terbukti

$$P III \quad n = k+1$$

diambil dari

$$k^3 = \left(\frac{k}{2} (k+1) \right)^2$$

$$k^3 + (k+1)^3 = \left(\frac{k}{2} (k+1) \right)^2 + (k+1)^3$$

$$= \frac{k^2}{4} (k+1)^2 + (k+1)^3$$

$$= (k+1)^2 \left(\frac{k^2}{4} + (k+1) \right)$$

$$= (k+1)^2 \left(\frac{k^2 + 4k + 4}{4} \right)$$

$$= \frac{(k+1)^2}{4} (k+2)^2$$

$$= \left(\frac{(k+1)}{2} ((k+1)+1) \right)^2$$

Terbukti

5. Tentukan kompleksitas dari algoritme berikut:

<pre> 1 int search(int data[], int size, int target) 2 { 3 int i; 4 for (i=0; i<size; i++) 5 { 6 if (data[i] == target) 7 return i; 8 } 9 return -1; 10 }</pre>	O(n)
<pre> int a = 0, b = 0; for (i = 0; i < N; i++) { a = a + rand(); } for (j = 0; j < M; j++) { b = b + rand(); }</pre>	O(n)
<pre> int a = 0; for (i = 0; i < N; i++) { for (j = N; j > i; j--) { a = a + i + j; } }</pre>	O(n ²)
<pre> int i, j, k = 0; for (i = n / 2; i <= n; i++) { for (j = 2; j <= n; j = j * 2) { k = k + n / 2; } }</pre>	O(n ²)

6. Efisiensi suatu algoritma dapat diukur dengan menghitung cost yang dilihat dari operasi operasi dasar yang dijalankan dalam suatu algoritme. Hitunglah total cost beberapa potongan program dibawah ini

a	<pre> min = 0 max = 0 while(n != 0){ scanf("%d",&num) if(num >= max) max = num else if(num <= min) min = num printf("%d %d\n",min,max) n = n - 1 } </pre>	<table> <tr> <td>c1</td><td>n</td><td rowspan="10"> Total cost : $C1 + c2 + c3*(n+1) + c4 + c5 + c6 + c7 + c8 + c9 + c10$ </td></tr> <tr><td>c2</td><td>n</td></tr> <tr><td>c3</td><td>n+1</td></tr> <tr><td>c4</td><td>n</td></tr> <tr><td>c5</td><td>n</td></tr> <tr><td>c6</td><td>n</td></tr> <tr><td>c7</td><td>n</td></tr> <tr><td>c8</td><td>n</td></tr> <tr><td>c9</td><td>n</td></tr> <tr><td>c10</td><td>n</td></tr> </table>	c1	n	Total cost : $C1 + c2 + c3*(n+1) + c4 + c5 + c6 + c7 + c8 + c9 + c10$	c2	n	c3	n+1	c4	n	c5	n	c6	n	c7	n	c8	n	c9	n	c10	n
c1	n	Total cost : $C1 + c2 + c3*(n+1) + c4 + c5 + c6 + c7 + c8 + c9 + c10$																					
c2	n																						
c3	n+1																						
c4	n																						
c5	n																						
c6	n																						
c7	n																						
c8	n																						
c9	n																						
c10	n																						
b	<pre> for(int i=0; i<n; i++){ for(int j=0; j<m; j++){ printf("%d",a[i][j]); if(j==m-1) printf("\n"); else printf(" "); } } </pre>	<table> <tr> <td>c1</td><td>n+1</td><td rowspan="6"> Total cost : $c1*(n+1) + c2*n*(m+1) + c3*nm + c4*nm + c5*nm + c6*nm$ </td></tr> <tr><td>c2</td><td>$n*(m+1)$</td></tr> <tr><td>c3</td><td>nm</td></tr> <tr><td>c4</td><td>nm</td></tr> <tr><td>c5</td><td>nm</td></tr> <tr><td>c6</td><td>nm</td></tr> </table>	c1	n+1	Total cost : $c1*(n+1) + c2*n*(m+1) + c3*nm + c4*nm + c5*nm + c6*nm$	c2	$n*(m+1)$	c3	nm	c4	nm	c5	nm	c6	nm								
c1	n+1	Total cost : $c1*(n+1) + c2*n*(m+1) + c3*nm + c4*nm + c5*nm + c6*nm$																					
c2	$n*(m+1)$																						
c3	nm																						
c4	nm																						
c5	nm																						
c6	nm																						

7. Tentukan nilai grow rate dari fungsi berikut jika memproses n data sebesar 5, 25, 125, 625,

Nilai	5	25	125	625
log(n)	2.32	4.64	6.96	9.28
n log(n)	4.64	9.28	13.93	18.57
\sqrt{n}	2.23	5	11.1803	25

n^3	125	15625	1953125	244140625
2^n	32	33554432	$4.25 \cdot 10^{37}$	$1.39 \cdot 10^{188}$
$n!$	120	$1.55 \cdot 10^{25}$	$1.88 \cdot 10^{209}$	Tak hingga