



Lembar Kerja Responsi 3
Mata Kuliah KOM 401 Analisis Algoritme
Semester Ganjil Tahun Akademik 2020/2021

Asisten Praktikum:

- 1. M. Faishal Wicaksono R.**
- 2. Zaki Muttaqin**

-
1. Diberikan tabel yang berisi kompleksitas waktu $T(n)$ yang digunakan oleh sebuah algoritme untuk menyelesaikan sebuah masalah berukuran n . Pilihlah suku yang paling dominan dan tentukan kompleksitas big-O dari setiap algoritme. Tuliskan pula urutan algoritme mulai dari kompleksitas yang paling rendah.

No	$T(n)$	Suku Dominan	$O(\dots)$	Urutan
1	$5 + 10n + n^3$			
2	$n^{1.75} + \log n^2 + n \log n$			
3	$n \log_3 n + n \log_2 n$			
4	$\log_2 \log_2 \log_2 n + (\log_2 n)^3$			
5	$10n! + n^{1000} + 10n$			
6	$\log_{10} n^{1000} + 99n^{10}$			
7	$100n + n^3 + 75$			
8	$100^n + n^{100} + 100$			
9	$2^n + 3^n + n^{3000}$			
10	$n! + 2^n + 15$			

2. Apakah pernyataan berikut benar? Jika salah, tuliskanlah pernyataan yang benar.

Pernyataan	Benar atau Salah	Perbaiki
Aturan penjumlahan: $O(f + g) = O(f) + O(g)$		
Aturan perkalian: $O(f \cdot g) = O(f) \cdot O(g)$		
Aturan Transivitas: jika $g = O(f)$ dan $h = O(f)$ maka $g = O(h)$		
$5n + 8n^2 + 1000n^3 = O(n^4)$		
$5n + 8n^2 + 100n^3 = O(n^2 \log n)$		

3. Algoritme A dan B memerlukan $T_A(n) = n^2 \log_{10} n$ dan $T_B(n) = 5n^2$ untuk suatu masalah dengan ukuran n . Pilihlah algoritme yang lebih baik berdasarkan konsep big-O. Carilah ukuran masalah n_0 sedemikian sehingga untuk $n > n_0$ salah satu algoritme pasti lebih baik daripada algoritme lainnya. Jika ukuran masalah $n \leq 10^9$, algoritme manakah yang lebih baik?
4. Dua buah algoritme A dan B digunakan untuk memproses sebuah pangkalan data yang besar, mengandung 10^{12} record. Algoritme memerlukan $T_A(n) = 2.5n$, sedangkan $T_B(n) = 0.1 n \log_2 n$. Manakah algoritme yang lebih baik sesuai dengan pengertian big-O? Kapan salah satu algoritme lebih baik daripada algoritme yang lain?
5. Tentukan kompleksitas dari potongan kode berikut:

```

for( i = n; i > 0; i /= 2){
    for( j = 1; j < n; j *=2 ){
        for( k = 0; k < n; k += 3){
            ... // operasi yang jumlahnya konstan
        }
    }
}

```

6. Dibawah ini terdapat algoritme X dan algoritme Y. Berapakah nilai n_0 sedemikian sehingga $f_x(n) > f_y(n)$ atau sebaliknya?

$f_x(n)$	$f_y(n)$	n_0
$n^2 + n$	$-n + 80$	
$2n^2 - 7$	$5n + 5$	
$2(n^2 - 2n)$	$n^2 + 21$	
$n(3n - 20n)$	$4n + 27$	
$(n - 8)^2$	16	

7. Berikut ini terdapat dua buah algoritme, algoritme tersebut memiliki suatu fungsi untuk melakukan suatu pekerjaan:

$$f_a(n) = 10\,000(n - 1000)$$

$$f_b(n) = \frac{1000\,000}{100n + 1}$$

- Berapakah nilai n_0 sedemikian sehingga $f_b(n) > f_a(n)$?
- Apabila anda memiliki ukuran data sebanyak 10 000, algoritme manakah yang lebih baik untuk digunakan?
- Misalkan algoritme tersebut akan dibuat menjadi program yang akan berjalan di komputer yang mampu mengeksekusi 10^8 instruksi dalam waktu satu detik. Berapakah waktu yang diperlukan oleh kedua algoritme tersebut (dalam satuan detik) untuk memproses sebanyak data yang ada pada tabel? Tuliskan jawaban Anda dalam tabel berikut:

n	Waktu Eksekusi Algoritme A	Waktu Eksekusi Algoritme b
2 000		
5 000		
10 000		

75 000		
1 000 000		

8. Manakah notasi asimtotik yang paling cocok digunakan untuk menunjukkan relasi di antara dua fungsi berikut ini:

No	$f(n)$	$g(n)$	$f(n) = \dots g(n)$
1.	1	99	
2.	n^k , k adalah konstanta > 1	c^n , c adalah konstanta > 1	
3.	$n^3 + 2n^2 + 10$	$7500n^2 + 89000n + 50$	
4.	$\log(n)$	$\log(n^k)$	
5.	$\sqrt{n} + 2$	$\log(n)$	
6.	$\log(n) + n$	$\log(n^2)$	
7.	$\sqrt{n+2}$	$\sqrt{n} + 2$	
8.	$n!$	$(n+1)!$	
9.	c^n , c adalah konstanta > 1	$n!$	
10.	n^n	$n!$	

9. Identifikasi kondisi yang menyebabkan *worst case*, *average case*, dan *best case* pada linear search berikut, dengan n adalah ukuran input dan x adalah target pencarian.

```

int search(int arr[], int n, int x)
{
    int i;
    for (i=0; i<n; i++)
    {
        if (arr[i] == x)
            return i;
    }
    return -1;
}

```

10. *Binary search tree* (BST) adalah salah satu bentuk struktur data yang menggambarkan hubungan hierarki antar elemen-elemennya dengan mengikuti aturan sebagai berikut.

- a. Semua data dibagian kiri sub-tree dari node t selalu lebih kecil dari data dalam node t itu sendiri.
- b. Semua data dibagian kanan sub-tree dari node t selalu lebih besar atau sama dengan data dalam node t.

Tentukan skenario *worst case* dan *best case* serta kompleksitasnya ketika anda melakukan pencarian suatu elemen pada BST.

11. Insertion sort merupakan algoritma pengurutan yang menghasilkan list terurut dengan mengurutkan satu elemen dalam satu waktu. Cara kerja insertion sort mirip seperti pengurutan sebuah deck kartu yang digambarkan oleh pseudocode dibawah ini.

```

INSERTION-SORT(A)
For  $j = 2$  to  $n$  DO
     $key = A[j]$ 
     $i = j - 1$ 
    WHILE  $i > 0$  and  $A[i] > key$  DO
         $A[i + 1] = A[i]$ 
         $i = i - 1$ 
    OD
     $A[i + 1] = key$ 
OD

```

Jelaskan secara singkat bagaimana algoritma ini bekerja berdasarkan pseudocode di atas dan lakukan analisis skenario yang dapat menyebabkan *worst case* dan *best case* pada algoritma ini.