

EXERCISE 1: KOMPLEKSITAS WAKTU

dikerjakan saat perkuliahan tatap muka

1. (Computing time complexity)

Hitunglah kompleksitas waktu kasus terbaik (*best-case*), kasus terburuk (*worst-case*), dan kasus rata-rata (*average-case*) dari algoritma berikut!

Algorithm 1 Sequential search

```

1: procedure SEQSEARCH( $A[1..n], x$ )
2:   found  $\leftarrow$  False
3:    $i \leftarrow 1$ 
4:   while (not found) and ( $i \leq N$ ) do
5:     if ( $A[i] = x$ ) then found  $\leftarrow$  True
6:     else  $i \leftarrow i + 1$ 
7:   end if
8: end while
9:   if (found) then index  $\leftarrow i$ 
10:  else index  $\leftarrow 0$ 
11:  end if
12: end procedure

```

2. (Dominant terms in a function)

Pilih suku yang paling signifikan (dominan) dan tentukan kompleksitas waktu \mathcal{O} terkecil dari setiap algoritma! Berikan bukti singkat atau penjelasan untuk setiap fungsi untuk mengklarifikasi jawaban Anda!

| | Expression | Dominant term(s) | $\mathcal{O}(\cdot)$ |
|-----|--|------------------|----------------------|
| 1. | $5 + 0.001n^3 + 0.025n$ | | |
| 2. | $500n + 100n^{1.5} + 50n \log_{10} n$ | | |
| 3. | $0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$ | | |
| 4. | $n^2 \log_2 n + n(\log_2 n)^2$ | | |
| 5. | $n \log_3 n + n \log_2 n$ | | |
| 6. | $3 \log_8 n + \log_2 \log_2 \log_2 n$ | | |
| 7. | $2n + n^{0.5} + 0.5n^{1.25}$ | | |
| 8. | $0.01n \log_2 n + n(\log_2 n)^2$ | | |
| 9. | $100n \log_3 n + n^3 + 100n$ | | |
| 10. | $0.003 \log_4 n + \log_2 \log_2 n$ | | |

3. (Comparing time complexities of two algorithms)

- (a) Suatu algoritma memiliki kompleksitas waktu $\mathcal{O}(f(n))$ dan *running time* $T(n) = cf(n)$. Jika algoritma tersebut membutuhkan waktu 10 detik untuk memproses 1000 data, berapakah waktu yang dibutuhkan untuk memproses 100,000 data jika $f(n) = n$? Bagaimana halnya jika $f(n) = n^3$? (Hint: tentukan terlebih dahulu nilai c dalam f)

- (b) Algoritma **A** dan **B** masing-masing memiliki kompleksitas waktu $T_A(n) = 5n \log_{10} n$ and $T_B(n) = 25n$ mikrodetik, jika diberikan input dengan ukuran n . Dalam aturan Big-O, algoritma manakah yang bekerja lebih baik? Berikan batasan nilai n agar algoritma **B** memiliki kompleksitas waktu yang lebih baik!
- (c) Suatu perangkat lunak **A** dan **B** masing-masing memiliki kompleksitas waktu $\mathcal{O}(n \log n)$ dan $\mathcal{O}(n^2)$. Keduanya memiliki running time $T_A(n) = c_A \cdot n \log_{10} n$ dan $T_B(n) = c_B \cdot n^2$ milidetik untuk memproses data dengan ukuran n . Selama proses pengujian, waktu rata-rata pemrosesan $n = 10^4$ data dengan perangkat **A** dan **B** masing-masing adalah 100 milidetik dan 500 detik.
- Berikan batasan nilai n agar perangkat **A** bekerja lebih cepat dibandingkan perangkat **B**, dan berikan batasan nilai n untuk kondisi sebaliknya.
 - Perangkat mana yang Anda rekomendasikan jika kita harus memproses data dengan ukuran tidak lebih dari $n = 10^9$?

(Hint: kerjakan seperti soal (a), perhatikan bahwa $f(n) = c \cdot T(n)$, yakni running time sebanding dengan kompleksitas waktu)