

LAPORAN PRAKTIKUM

WORKSHEET 4



Oleh:

Muhamad Ilham Habib (140810180018)

Kelas B

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

JATINANGOR

2020

I. Tujuan

1. Mahasiswa paham tentang paradigma divide & conquer
2. Mahasiswa paham tentang rekurensi
3. Mahasiswa dapat membedah algoritma sorting dengan paradigma divide & conquer

II. Landasan Teori

Paradigma divide & conquer

Divide & Conquer merupakan teknik algoritmik dengan cara memecah input menjadi beberapa bagian, memecahkan masalah di setiap bagian secara rekursif, dan kemudian menggabungkan solusi untuk subproblem ini menjadi solusi keseluruhan. Menganalisis running time dari algoritma divide & conquer umumnya melibatkan penyelesaian rekurensi yang membatasi running time secara rekursif pada instance yang lebih kecil.

Pengenalan rekurensi

- a. Rekurensi adalah persamaan atau ketidaksetaraan yang menggambarkan fungsi terkait nilainya pada input yang lebih kecil. Ini adalah fungsi yang diekspresikan secara rekursif
- b. Ketika suatu algoritma berisi panggilan rekursif untuk dirinya sendiri, running time-nya sering dapat dijelaskan dengan perulangan
- c. Sebagai contoh, running time worst case $T(n)$ dari algoritma merge-sort dapat dideskripsikan dengan perulangan :

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \end{cases}$$

with solution $T(n) = \Theta(n \lg n)$.

III. Worksheet 4

Studi Kasus 1: MERGE SORT

Setelah Anda mengetahui Algoritma Merge-Sort mengadopsi paradigma divide & conquer, lakukan Hal berikut:

1. Buat program Merge-Sort dengan bahasa C++
2. Kompleksitas waktu algoritma merge sort adalah $O(n \lg n)$. Cari tahu kecepatan komputer Anda dalam memproses program. Hitung berapa running time yang dibutuhkan apabila input untuk merge sort-nya adalah 20?

Jawaban

1. Ada di luar file
2. Dengan spesifikasi komputer saya yaitu seperti ini :

System

Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Installed memory (RAM): 4.00 GB (3.88 GB usable)
System type: 64-bit Operating System, x64-based processor
Pen and Touch: No Pen or Touch Input is available for this Display

dengan input-an sebanyak 20, Running time yang didapatkan dalam sorting merge sortnya adalah:

```
Banyak data yang akan diurutkan: 20
Masukkan elemen ke-1: 77
Masukkan elemen ke-2: 66
Masukkan elemen ke-3: 2
Masukkan elemen ke-4: 11
Masukkan elemen ke-5: 1
Masukkan elemen ke-6: 31
Masukkan elemen ke-7: 4
Masukkan elemen ke-8: 71
Masukkan elemen ke-9: 5
Masukkan elemen ke-10: 99
Masukkan elemen ke-11: 6
Masukkan elemen ke-12: 81
Masukkan elemen ke-13: 7
Masukkan elemen ke-14: 97
Masukkan elemen ke-15: 8
Masukkan elemen ke-16: 75
Masukkan elemen ke-17: 9
Masukkan elemen ke-18: 198
Masukkan elemen ke-19: 10
Masukkan elemen ke-20: 0

Data Terurut: |0|1|2|4|5|6|7|8|9|10|11|31|66|71|75|77|81|97|99|198
Process returned 0 (0x0)   execution time : 45.887 s
Press any key to continue.
```

Jawaban :

Studi Kasus 2: SELECTION SORT

Selection sort merupakan salah satu algoritma sorting yang berparadigma divide & conquer. Untuk membedah algoritma selection sort, lakukan langkah-langkah berikut:

- Pelajari cara kerja algoritma selection sort
- Tentukan $T(n)$ dari rekurensi (pengulangan) selection sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

- Selesaikan persamaan rekurensi $T(n)$ dengan **metode recursion-tree** untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ
- Lakukan implementasi koding program untuk algoritma selection sort dengan menggunakan bahasa C++

- Pelajari cara kerja algoritma selection sort
- Tentukan $T(n)$ dari rekurensi (pengulangan) selection sort berdasarkan penentuan rekurensi divide & conquer:

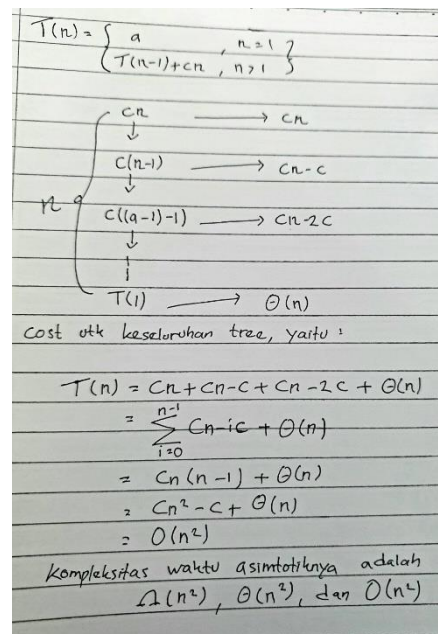
$$T(n) = \begin{cases} \theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

$T(n)$ dari rekurensi selection sort:

$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$

- Selesaikan persamaan rekurensi $T(n)$ dengan **metode recursion-tree** untuk mendapatkan kompleksitas waktu asimtotiknya dalam Big-O, Big-Ω, dan Big-Θ

metode recursion-tree



$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$

$$T(n) = cn + cn - c + cn - 2c + \dots + \Theta(n)$$

$$= \sum_{i=0}^{n-1} cn - ic + \Theta(n)$$

$$= cn(n-1) + \Theta(n)$$

$$= cn^2 - cn + \Theta(n)$$

$$= O(n^2)$$

Kompleksitas waktu asimtotiknya adalah $\Omega(n^2)$, $\Theta(n^2)$, dan $O(n^2)$

- Lakukan implementasi koding program untuk algoritma selection sort dengan menggunakan bahasa C++

Ada diluar file

Studi Kasus 3: INSERTION SORT

Insertion sort merupakan salah satu algoritma sorting yang berparadigma divide & conquer. Untuk membedah algoritma insertion sort, lakukan langkah-langkah berikut:

- Pelajari cara kerja algoritma insertion sort
- Tentukan $T(n)$ dari rekurensi (pengulangan) insertion sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

$T(n)$ dari rekurensi insertion sort:

$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$

- Selesaikan persamaan rekurensi $T(n)$ dengan **metode substitusi** untuk mendapatkan

kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ

insertion sort - metode substitusi

$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$
$$\begin{aligned} T(n) &= cn + T(n-1) \\ &= cn + \{c(n-1) + T(n-2)\} \\ &= cn + c(n-1) + \{c(n-2) + T(n-3)\} \\ &= \dots \\ &= cn + c(n-1) + c(n-2) + \dots + c + T(1) \\ &= c\{n + (n-1) + (n-2) + \dots + 2\} + a \\ &= c\{n(n-1)(n-2)/2\} + a \\ &= cn^2/2 + cn/2 + (a-c) \\ &= O(n^2) \end{aligned}$$

Maka kompleksitas waktu asimptotiknya adalah $\Omega(n^2)$, $\Theta(n^2)$, dan $O(n^2)$.

Studi Kasus 4: BUBBLE SORT

Bubble sort merupakan salah satu algoritma sorting yang berparadigma divide & conquer. Untuk membedah algoritma bubble sort, lakukan langkah-langkah berikut:

- Pelajari cara kerja algoritma bubble sort
- Tentukan $T(n)$ dari rekurensi (pengulangan) insertion sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

$T(n)$ dari rekurensi Bubble sort:

$$T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$$

- Selesaikan persamaan rekurensi $T(n)$ dengan **metode master** untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ
- Lakukan implementasi koding program untuk algoritma bubble sort dengan menggunakan bahasa C++

Metode master

Karena $T(n) = \begin{cases} a & , n = 1 \\ T(n-1) + cn & , n > 1 \end{cases}$, sehingga $a = 1$ dan $b = 1$. yang artinya tidak

memenuhi syarat untuk metode master yaitu ($a \geq 1$ dan $b > 1$), maka rekurensi algoritma bubble sort tidak bisa menggunakan metode master