

# Ujian Tengah Semester

Jurusan Sistem Informasi, Tahun Akademik 2017/2018

Mata kuliah	BS204 / Algoritma	Pengesahan
Semester	Genap	Kaprodi
Sifat Ujian	Closed Book	
Hari / Tanggal	Rabu / 07.03.2018	
Waktu	120 menit	
Soal	Tidak Dibawa Pulang	(_____)
Pengajar	Setia Budi	NIK:_____

Kerjakan soal-soal berikut secara berurutan.

Soal ujian ini disusun dengan mengacu pada Grokking Algorithms chapter 01 - chapter 06.

Setiap soal memiliki bobot nilai 4.

## CHAPTER 01

1. Anda diminta untuk menebak sebuah angka dari sekumpulan angka terurut sejumlah 150 angka. Anda sudah menebak 3 kali dengan *Binary Search* dan gagal; dua tebakan pertama terlalu kecil dan tebakan ketiga terlalu besar. Untuk tebakan keempat, di manakah posisi index low, mid, dan high? Sertakan ilustrasi gambar untuk memperjelas jawaban!

[Chapter 01]

- tebakan 1: lo=0, mi=74, hi=149
- tebakan 2: lo=75, mi=112, hi=149
- tebakan 3: lo=113, mi=131, hi=149
- tebakan 4: lo=113, mi=121, hi=130

2. Anda diminta melakukan pencarian data dari sekumpulan data berjumlah 512 element. Sebutkan jumlah langkah terbanyak yang dibutuhkan untuk melakukan pencarian dengan *simple search* dan *binary search*! Jelaskan!

[Chapter 01]

- Simple search memiliki running time  $O(n)$  sehingga membutuhkan 512 langkah.
- Binary search memiliki running time  $O(\log n)$  sehingga membutuhkan 9 langkah.

3. Jelaskan apa yang dimaksud dengan *Big O* notation!

[Chapter 01]

Big O adalah notasi yang digunakan untuk mendeskripsikan kecepatan dari suatu algoritma terhadap ukuran data yang ditanganinya. Secara umum Big O akan mengadopsi kondisi terburuk dalam mendeskripsikan kecepatan algoritma.

4. Anda diminta memilih salah satu dari dua algoritma untuk menyelesaikan permasalahan yang sama. Algoritma pertama memiliki running time *linear* dan algoritma kedua memiliki running time *logaritmik*. Algoritma manakah yang akan Anda pilih dengan mempertimbangkan efisiensi kecepatan? Beri penjelasan!

[Chapter 01]

Algoritma dengan running time logaritmik akan jauh lebih cepat dibandingkan algoritma dengan running time linear. Untuk menangani data berukuran 512, algoritma dengan running time linear akan membutuhkan 512 langkah sedangkan algoritma dengan running time logaritmik hanya membutuhkan 9 langkah untuk menyelesaikan permasalahan yang sama.

---

## CHAPTER 02

---

5. Jelaskan perbedaan mendasar antara *array* dan *linked-list* terkait mekanisme penyimpanan data di memory!

[Chapter 02]

Array akan menyimpan sekumpulan data di memory tepat bersebelahan antara data yang satu dengan lainnya. Sedangkan linked-list dapat menyimpan data di mana pun di memory. Hanya saja untuk setiap data yang tersimpan, linked-list perlu menyimpan informasi alamat memory dari data selanjutnya.

6. Sebut dan jelaskan dua mekanisme untuk mengakses data yang tersimpan di dalam memory!

[Chapter 02]

- Sequential access: Adalah mekanisme akses data secara terurut di mana data ke n hanya dapat diakses setelah melakukan akses terhadap data ke n-1. Hal ini dikarenakan alamat dari data ke n disimpan pada data ke n-1. Mekanisme semacam ini didapati pada linked-list.
- Random access: Adalah mekanisme akses data secara acak di mana data ke n dapat diakses kapan pun tanpa perlu melakukan akses terhadap data sebelumnya. Hal ini dimungkinkan karena sekumpulan data disimpan tepat bersebelahan. Mekanisme semacam ini ditemui pada array.

7. Sebut dan jelaskan kondisi terburuk dari penerapan array terkait penambahan data baru!

[Chapter 02]

Kondisi terburuk dari array terkait penambahan data baru adalah ketika alamat memory untuk menempatkan data baru tersebut sudah ditempati data lain. Untuk mengatasi kondisi ini, akan dilakukan pencarian alokasi memory baru yang dapat digunakan untuk menampung keseluruhan data array, termasuk data array yang baru. Setelah ditemukan, seluruh data array akan dipindahkan dari alokasi memory yang lama ke alokasi memory yang baru.

---

## CHAPTER 03

---

8. Sebut dan jelaskan dua komponen utama pada recursion!

[Chapter 03]

- Recursive case: adalah bagian ketika sebuah function memanggil dirinya sendiri sehingga memungkinkan terjadinya proses perulangan atau looping.
- Base case: adalah kondisi yang menjadikan sebuah recursion berhenti memanggil dirinya sendiri. Base case dibutuhkan untuk mencegah terjadinya infinite-loop.

**9. Sebut dan jelaskan dua operasi dasar pada *stack*!**

[Chapter 03]

- Push: adalah proses penambahan data pada sebuah stack, di mana data baru akan ditempatkan di posisi paling atas dari stack.
- Pop: adalah proses pengambilan data dari suatu stack, di mana data yang diambil adalah data yang berada pada posisi paling atas.

**10. Jelaskan apa yang dimaksud dengan *call stack*!**

[Chapter 03]

Call stack adalah data structure yang digunakan untuk menangani setiap pemanggilan function.

---

CHAPTER 04

---

**11. Bila anda memiliki lahan berukuran 1680 meter x 1024 meter, berapakah ukuran tiap petak yang dihasilkan dengan menerapkan *Algoritma Euclid* untuk membagi lahan tersebut? Sertakan ilustrasi gambar untuk memperjelas jawaban:**

[Chapter 04]

- $\text{GCF}(1680, 1024) = 16$
- $1680 \div 1024 = 1 \text{ R } 656$  ( $1680 = 1 \times 1024 + 656$ )
- $1024 \div 656 = 1 \text{ R } 368$  ( $1024 = 1 \times 656 + 368$ )
- $656 \div 368 = 1 \text{ R } 288$  ( $656 = 1 \times 368 + 288$ )
- $368 \div 288 = 1 \text{ R } 80$  ( $368 = 1 \times 288 + 80$ )
- $288 \div 80 = 3 \text{ R } 48$  ( $288 = 3 \times 80 + 48$ )
- $80 \div 48 = 1 \text{ R } 32$  ( $80 = 1 \times 48 + 32$ )
- $48 \div 32 = 1 \text{ R } 16$  ( $48 = 1 \times 32 + 16$ )
- $32 \div 16 = 2 \text{ R } 0$  ( $32 = 2 \times 16 + 0$ )

**12. Sebutkan salah satu karakteristik dari *functional programming*:**

[Chapter 04]

Bahasa pemrograman yang menerapkan functional programming tidak memiliki kemampuan untuk melakukan looping secara eksplisit. Looping dalam functional programming akan bergantung pada recursion.

**13. Urutkan kumpulan data berikut ini menggunakan *Quicksort* dengan menggunakan data paling *kiri* sebagai *pivot*! Sertakan ilustrasi gambar untuk memperjelas jawaban!**

6	2	7	4	3	5	2	0	9	8	1	3	5	2	8
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

[Chapter 04]

left pivot

14. Urutkan kumpulan data berikut ini menggunakan *Quicksort* dengan menggunakan data paling *kanan* sebagai *pivot*! Sertakan ilustrasi gambar untuk memperjelas jawaban!

6	2	7	4	3	5	2	0	9	8	1	3	5	2	8
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

[Chapter 04]

right pivot

15. Urutkan kumpulan data berikut ini menggunakan *Quicksort* dengan menggunakan data posisi *tengah* sebagai *pivot*! Sertakan ilustrasi gambar untuk memperjelas jawaban!

6	2	7	4	3	5	2	0	9	8	1	3	5	2	8
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

[Chapter 04]

middle pivot

---

## CHAPTER 05

---

16. Jelaskan apa yang dimaksud dengan *Hash Function*!

[Chapter 05]

Hash function adalah function yang digunakan untuk melakukan mapping dari suatu string (key) ke suatu nilai numerik (alokasi memory) tertentu

17. Sebutkan dua syarat utama *hash function*!

[Chapter 05]

- Konsisten dalam melakukan mapping suatu key ke value tertentu.
- Setiap key yang berbeda akan selalu dimapping ke value yang berbeda.

18. Jelaskan apakah yang dimaksud dengan *hash table*!

[Chapter 05]

Hash table adalah data structure yang menggabungkan penggunaan array dan hash function.

19. Jelaskan apakah yang dimaksud dengan *colission* pada *hash table*!

[Chapter 05]

Colission pada hash table adalah kondisi di mana dua atau lebih keys ditranslasikan/dimapping oleh suatu hash function pada alokasi slot memory yang sama.

**20. Jelaskan apakah yang dimaksud dengan *load factor* pada *hash table*!**

[Chapter 05]

Load factor adalah perbandingan antara jumlah alokasi memory yang sudah terpakai terhadap total jumlah alokasi memory yang tersedia. Pada hash table, load factor 0.7 umum digunakan sebagai treshold untuk melakukan resizing.

---

CHAPTER 06

---

**21. Jelaskan perbedaan antara *directed graph* dan *undirected graph*!**

[Chapter 06]

Directed graph digunakan untuk memetakan hubungan satu arah sedangkan undirected digunakan untuk memetakan hubungan dua arah.

**22. Sebutkan dua jenis pertanyaan yang dapat dijawab dengan menerapkan *Breadth-first search*!**

[Chapter 06]

- Dalam suatu graph, apakah node A terhubung dengan node B?
- Bila node A terhubung dengan node B, manakah jalur terpendek yang menghubungkan node A ke node B?

**23. Sebut dan jelaskan operasi dasar pada *Queue*!**

[Chapter 06]

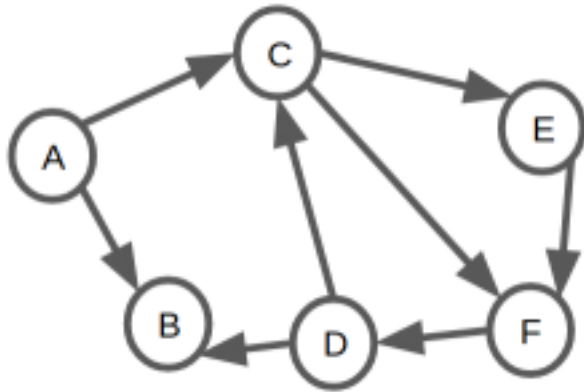
- Enqueue: operasi penambahan data pada queue di mana data ditambahkan pada bagian belakang queue.
- Dequeue: operasi pengambilan data dari suatu queue, di mana data terdepan yang akan diambil.

**24. Jelaskan perbedaan antara LIFO dan FIFO!**

[Chapter 06]

- LIFO: Last In First Out, adalah mekanisme kerja pada data structure di mana data yang terakhir kali dimasukkan akan menjadi data yang pertama kali dikeluarkan.
- FIFO: First In First Out, adalah mekanisme kerja pada data structure di mana data yang pertama kali dimasukkan akan menjadi data yang pertama kali dikeluarkan.

25. Sebutkan seluruh node beserta neighbours dari tiap node pada graph berikut ini:



[Chapter 06]

- A: [C, B]
- B: []
- C: [E, F]
- D: [C, B]
- E: [F]
- F: [D]