

Ujian Tengah Semester **IF2211 Strategi Algoritma**

Rabu, 26 Maret 2024

Waktu: 110 menit

Dosen: Rinaldi Munir (K1), Nur Ulfa Maulidevi (K2), Rila Mandala (K3), Monterico Adrian (K3)

---

*Berdoalah terlebih dahulu agar Anda berhasil dalam mengerjakan ujian ini!*

**Algoritma Greedy**

1. Dalam sebuah permainan *adventure*, pemain dapat membawa item *healing* seberat 50 satuan. Terdapat 4 item untuk *healing*: Elixir yang mempunyai berat 28 dan kemampuan *healing* 40, Potion yang mempunyai berat 22 dan kemampuan *healing* 32, Tonic yang mempunyai berat 24 dan kemampuan *healing* 35, dan Ointment yang mempunyai berat 32 dan kemampuan *healing* 45. Jika pemain hanya dapat membawa satu untuk tiap jenis item dan item dapat dibawa secara sebagian, tentukan item mana saja yang dibawa yang akan memberikan keuntungan maksimum dengan algoritma Greedy. **(Nilai: 15)**

**Brute Force & Divide and Conquer**

2. Diberikan sebuah larik dengan  $n$  elemen yang berisi nilai-nilai dalam bilangan riil. Kita akan mencari perbedaan maksimum antara dua elemen di dalam larik. Contoh:  $A = [4.5, 10, -2, 3.14, -7.25]$ , perbedaan maksimum adalah 17.25. **(Nilai: 15)**
  - (a) Jika diselesaikan secara *brute force*, bagaimana algoritmanya dan berapa kompleksitas waktunya?
  - (b) Jika diselesaikan secara *divide and conquer*, tuliskan langkah-langkahnya, tuliskan pseudo-code nya, tentukan kompleksitas waktunya dalam relasi rekurens, lalu notasi kompleksitas asimptotiknya.

**Teorema Master**

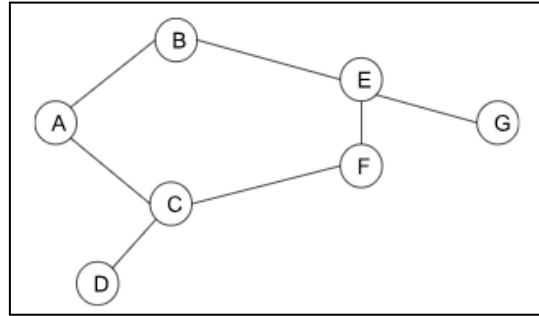
3. Dengan menggunakan Teorema Master, tentukan notasi notasi Big-Oh untuk kompleksitas waktu dalam bentuk relasi rekurens berikut. Jika teorema master tidak bisa diterapkan jelaskan alasannya: **(Nilai: 10)**
  - a.  $T(n) = 5T(n/4) + \sqrt{n}$
  - b.  $T(n) = T(\sqrt{n}) + 5$  (Petunjuk: misalkan  $n = 2^m$ )
  - c.  $T(n) = 2T(\sqrt{n}) + 1$  (Petunjuk: misalkan  $n = 2^m$ )

**Decrease and Conquer**

4. Terdapat sebuah larik bilangan bulat  $[57, 17, 4, 67, 98, 36, 100, 77, 82]$ . Tanpa melakukan pengurutan larik terlebih dulu, tentukan bilangan terkecil ke-3 dari larik tersebut dengan pendekatan *decrease and conquer by variable size* sesuai salindia kuliah. Catatan: bilangan terkecil ke-1 pada larik tersebut adalah 4, dan bilangan terkecil ke-9 pada larik adalah 100. **(Nilai 15)**
  - a. Tuliskan tahapan secara global proses pencarian bilangan terkecil ke-3 pada sembarang larik bilangan integer dengan ukuran sembarang dan tidak terurut, menggunakan *decrease and conquer*.
  - b. Tuliskan dengan lengkap iterasi sesuai jawaban (a) untuk larik pada soal. Satu iterasi adalah hingga partisi dihasilkan.

## DFS, BFS, IDS

5. Terdapat graf sebagai berikut.



Setiap simpul akan diberi sebuah warna, dan hanya dua warna yang digunakan yaitu merah dan hijau. Pewarnaan dimulai dari simpul A dengan warna merah. Penelusuran (traversal) harus digambarkan sebagai pohon. Sebuah simpul akan diberi warna yang berbeda dengan simpul *parent* pada pohon penelusuran. Jika sebuah simpul memiliki lebih dari satu simpul tetangga, maka yang diperiksa terlebih dahulu sesuai urutan abjad. **(Nilai 20)**

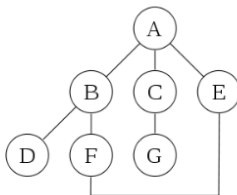
a. Tuliskan urutan pemberian warna simpul jika menggunakan pendekatan **Breadth First Search (BFS)**, dengan menggambarkan traversal sebagai pohon BFS. Tuliskan warna setiap simpul di sebelah simpul. Contoh:



b. Tuliskan urutan pemberian warna simpul jika menggunakan pendekatan **Depth First Search (DFS)**, dengan menggambarkan traversal sebagai pohon DFS. Tuliskan warna setiap simpul di sebelah simpul, seperti pada soal (a).

c. Berdasarkan jawaban soal (a) dan (b), apakah graf tersebut bipartit? Jelaskan dengan singkat.

6. Diketahui graf di bawah ini :



Tentukanlah bagaimana urutan bekerja dari algoritma IDS (*Iterative Deepening Search*) jika ditelusuri maksimum sampai level kedalaman 3. **(Nilai: 10)**

## Kapita Selekt

7. Tentukanlah apakah pernyataan di bawah ini salah atau benar, dan kemukakan alasannya :

- Strategi algoritmik Divide and Conquer berusaha untuk mencari solusi yang optimum.
- Strategi algoritmik Greedy tidak berusaha mencari solusi optimum karena ada kemungkinan untuk terjebak dalam optimum lokal.
- Dalam strategi algoritmik IDS (*iterative deepening search*), jika  $b$  adalah *branching factor* dan  $l$  adalah kedalaman maksimum dari Solusi pertamanya maka space complexity nya adalah  $O(b^l)$

**(Nilai: 15)**