

Ujian Tengah Semester **IF2211 Strategi Algoritma**

Rabu, 8 Maret 2023

Waktu: 120 menit

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

Berdoalah terlebih dahulu agar Anda berhasil dalam mengerjakan ujian ini!

Brute force dan Greedy Algorithm

1. Diberikan dua buah larik integer, $A[1..n]$ dan $B[1..n]$. Kita ingin meminimumkan $\sum_{i=1}^n A[i] * B[i]$.
 - (a) Jika diselesaikan secara *brute force*, bagaimana caranya dan berapa kompleksitas algoritmanya?
 - (b) Jika diselesaikan dengan algoritma *greedy*, bagaimana strateginya? Berapa kompleksitasnya? Ilustrasikan jawaban anda dengan contoh berikut: $A = [7, 5, 1, 4]$ dan $B = [6, 17, 9, 3]$.

(Nilai: 10)

2. Terdapat n buah mata kuliah yang akan dijadwalkan pada sejumlah ruang kuliah. Setiap mata kuliah i memiliki waktu mulai s_i dan waktu selesai f_i . Bagaimana menjadwalkan semua kuliah pada ruang-ruang kuliah sehingga jumlah ruang kuliah yang dipakai seminimal mungkin? Tidak boleh ada dua atau lebih kuliah yang bentrok waktunya (beririsan waktunya) menggunakan ruang kuliah yang sama. Jelaskan strategi *greedy*-nya seperti apa dan berapa kompleksitas waktunya. Ilustrasikan jawaban anda dengan contoh berikut:

(Nilai: 15)

$$n = 10, (s_i, f_i) = [(5, 6), (4, 7), (2, 5), (1, 4), (3, 7), (8, 10), (7, 8), (1, 3), (6, 9), (5, 8)]$$

Teorema Master

3. Tentukan notasi notasi Big-Oh untuk kompleksitas waktu dalam bentuk relasi rekurens berikut. Jika teorema master tidak bisa diterapkan jelaskan alasannya:
 - a. $T(n) = 27T(n/3) + n^3$
 - b. $T(n) = 0.5T(n/4) + n/2$
 - c. $T(n) = 16T(n/2) - n^2$

(Nilai: 5)

Divide and Conquer

4. Di bawah ini adalah algoritma dengan strategi *divide and conquer* untuk menghitung jumlah daun dari sebuah pohon biner

```
Algoritma LeafCounter( T )  
//Menghitung secara rekursif jumlah daun dalam sebuah pohon biner.  
//Input: sebuah pohon biner T  
//Output: Jumlah daun dalam pohon T  
if  $T = \emptyset$   
    return 0  
else  
    return LeafCounter(TL)+ LeafCounter (TR)
```

- a. Apakah algoritma di atas benar atau salah (bukan secara sintaksis) untuk semua kemungkinan pohon biner ? Jika anda menjawab benar, maka jelaskan mengapa algoritma di atas benar. Jika anda menjawab salah, maka berikan sebuah contoh data (berupa pohon biner) dimana algoritma itu akan memberikan jawaban yang salah. **(Nilai : 5)**

- b. Berdasarkan notasi algoritma di atas, buatlah dengan notasi di atas, algoritma untuk menghitung jumlah level yang ada pada sebuah pohon biner sbb (**Nilai : 5**) :

Algoritma Levels(T)

//Input: pohon biner T

//Output: jumlah level pohon biner T

Decrease and Conquer

5. Diberikan sebuah array yang dimulai dengan indeks 0 yang isinya adalah n bilangan integer yang sudah terurut dan perbedaan diantara 2 bilangan yang berurutan adalah selalu konstan. Diketahui bahwa diantara n bilangan tersebut ada 1 buah bilangan yang sengaja dihilangkan. Carilah sebuah bilangan yang hilang dalam urutan n bilangan tersebut.

Sebagai contoh jika diberikan array sbb : `int nums[] = { 5, 7, 9, 11, 15 }`; maka *missing* elemennya adalah 13.

- a. Selesaikan persoalan di atas (boleh *pseudo-code* atau memakai kata-kata sehari-hari namun harus jelas) dengan menggunakan strategi algoritma Brute Force dan hitunglah kompleksitas waktunya. (**Nilai : 5**)
- b. Selesaikan algoritma tersebut dengan menggunakan pseudo-code (C-like) dengan menggunakan algoritma *Decrease and Conquer* yang harus lebih efisien daripada metode brute force di bagian **a** (harus ada komentar untuk setiap instruksinya, di instruksi itu melakukan apa) dengan membuat body dari fungsi di bawah ini :

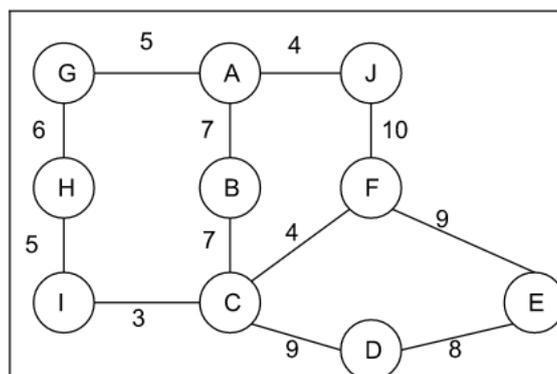
```
int findMissingTerm(int nums[], int n)
{
}
```

Dimana `nums[]` adalah array yang sudah berisi bilangan integer, dan n adalah jumlah bilangan yang ada di array `nums[]`. (**Nilai : 10**)

- c. Jalankan step-step dari algoritma yang sudah anda buat di bagian **b** pada array { 5, 7, 9, 11, 15 } (**Nilai : 5**)
- d. Tentukan kompleksitas waktu algoritma yang anda buat tersebut. (**Nilai : 5**)

Brute Force, DFS dan BFS

Terdapat graf sebagai berikut. Simpul merepresentasikan kota, dan sisi merepresentasikan jalan yang menghubungkan antar kota dan jaraknya. **Prioritas pemilihan simpul berdasarkan urutan abjad.**



Gambar 1. Graf Terhubung

6. Jika ingin dicari jalur terpendek dari A menuju ke I. **(Nilai 10)**
 - a. Jelaskan langkah-langkah untuk mendapatkan jalur terpendek dengan memanfaatkan Brute Force.
 - b. Jelaskan dengan singkat kompleksitas waktu untuk jawaban butir (a) dalam Big O.

7. Ingin ditelusuri semua simpul pada gambar 1 menggunakan pendekatan BFS dan DFS. **(Nilai 10)**
 - a. Tuliskan urutan penelusuran semua simpul pada Gambar 1 yang dimulai dari simpul A dengan BFS.
 - b. Tuliskan urutan penelusuran semua simpul pada Gambar 1 yang dimulai dari simpul A dengan DFS.

8. Jika ingin dicari jalur dari A menuju ke I menggunakan BFS dan DFS. **(Nilai 15)**
 - a. Tuliskan proses pencarian jalur menggunakan BFS menggunakan contoh tabel 1.
 - b. Tuliskan proses pencarian jalur menggunakan DFS menggunakan contoh tabel 1.
 - c. Bandingkan jalur hasil pencarian (jarak) dari A menuju I menggunakan Brute Force, BFS, dan DFS.

Catatan: Untuk BFS dan DFS, simpul yang sudah pernah di ekspan tidak diperiksa lagi. Pencarian berhenti ketika simpul yang diperiksa adalah simpul goal (simpul I).

Tabel 1. Proses Pencarian Jalur dari Simpul A ke Simpul I

Iterasi	Simpul Ekspan	Simpul Hidup
1	A	B _A , G _A , J _A
2
...
Jalur Hasil:	...	
Jarak:	...	