

Ujian Tengah Semester IF2211 Strategi Algoritma Sem 2 2020-2021

UTS IF2211 dilaksanakan dengan waktu 100 menit (11.00 - 12.40 WIB). Ujian bersifat individu dan tutup buku.

Terdapat tiga bagian dalam ujian ini. Bagian pertama adalah identitas dan pernyataan kejujuran, bagian kedua adalah soal isian (essay), dan bagian ketiga adalah soal pilihan. Pastikan anda 'submit' hasil pekerjaan setelah menyelesaikan UTS.

*** Wajib**

1. Alamat email *

2. Nama *

3. NIM *

4. Kelas *

Tandai satu oval saja.

K1

K2

K3

K4

5. Tulis ulang pernyataan berikut: "Saya menyatakan bahwa saya mengerjakan UTS ini dengan sejujur-jujurnya, tanpa bantuan orang lain dan tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui saya mengerjakan UTS ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF2211 Semester 2 2020/2021. " *

Soal Essay

6. Diberikan sebuah himpunan P berisi n buah titik di suatu sistem koordinat 2 dimensi, dan sebuah integer $k \leq n$. Carilah himpunan C yang berisi k buah titik yang merupakan subset dari himpunan P , yang meminimalkan jarak maksimum dari semua titik yang ada di himpunan P ke titik terdekat di himpunan C .
Pertanyaan a) Carilah strategi Greedy yang akan memberikan solusi paling baik (bukan strategy yang "bodoh").
Pertanyaan b) Berikan contoh jalannya algoritma anda tersebut dengan memberikan contoh himpunan P , C , n , dan k , serta tahapan-tahapan dan hasil akhir yang didapatkan.
Pertanyaan c) hitunglah kompleksitas waktunya dengan menggunakan notasi Big-O.

7. (Selection Problem - Decrease and Conquer) Diberikan sebuah larik A berukuran $n = 9$ elemen. Elemen-elemen larik A adalah sebagai berikut: [4 1 10 9 7 12 8 2 15]. Kita akan mencari elemen terbesar di dalam larik tersebut dengan metode decrease and conquer (memanfaatkan algoritma partisi di dalam Quicksort versi 2). Tuliskan susunan elemen-elemen larik kondisi terakhir setelah ditemukan elemen terbesar. Format jawaban: [a1 a2 a3 a4 a5 a6 a7 a8 a9]. Contoh jawaban: [4 9 5 12 15 1 2 10 7]

[2 1 4 8 7 9 10 12 15]

8. Selesaikan persoalan berikut:

Sebuah larik A berisi deretan bilangan integer yang tidak terurut. Tuliskan algoritma dengan pendekatan **Divide and Conquer** dalam bentuk rekursif untuk mencari banyaknya elemen yang memiliki rentang nilai antara (dan termasuk) min dan mak. Sebagai contoh, jika $A = 10, 29, 89, 50, 34, 91, 39, 66, 20$ dengan nilai yang dicari dalam rentang antara $\text{min}=20$ dan $\text{mak}=40$, maka banyak elemen dengan rentang tersebut adalah 4 (empat). Lalu tuliskan analisis algoritma untuk mendapatkan kompleksitas dari perbandingan elemen yang dilakukan.

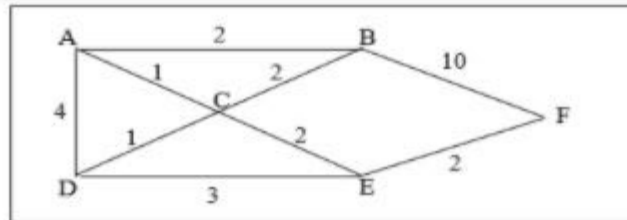
Untuk algoritma, gunakan nama dan urutan parameter masukan berikut:

CountRange (Input A : larik integer;
 $low, high$: integer; // batas indeks bawah dan atas larik A.
 min, mak : integer). // batas bawah dan atas rentang nilai yang dihitung.

Output CountRange adalah banyaknya elemen antara (dan termasuk) min dan mak.

Dalam penulisan untuk penurunan kompleksitas gunakan symbol '^' untuk menyatakan pangkat, sebagai contoh: "n pangkat 2" ditulis dengan "n^2".

9. Terdapat sebuah graf tidak berarah seperti pada Gambar 1 berikut ini. Simpul merepresentasikan kota, dan bilangan yang terdapat pada sisi adalah jarak antara dua kota. Persoalan yang ingin diselesaikan adalah mencari jalur dari kota A ke kota F, dan diharapkan dapat memberikan jalur dengan jarak total minimum. (i) Jika digunakan pendekatan Brute Force, tuliskan langkah penyelesaiannya secara garis besar, kemudian tuliskan jalur hasil penyelesaiannya serta jarak total jalur hasil tersebut. (ii) Jika digunakan pendekatan BFS, tuliskan proses pencarian dengan cara seperti pada gambar. (iii) Jika digunakan pendekatan DFS, tuliskan proses pencarian dengan cara seperti pada gambar. Catatan: untuk pencarian dengan BFS atau DFS, urutan prioritas simpul sesuai dengan urutan abjad. Perhatikan urutan penulisan simpul hidup jika terdapat simpul baru yang ditambahkan pada daftar simpul hidup, sesuai dengan teknik BFS atau DFS.



Gambar 1.

BFS

Iterasi 1:

Simpul Ekspan (Simpul yang diperiksa) = A

Simpul Hidup = B[A] C[A] D[A] (catatan: C[AB] artinya simpul C dari jalur AB)

Iterasi 2:

Simpul Ekspan (Simpul yang diperiksa) = [tuliskan simpul yang diperiksa berikutnya]

Simpul Hidup = [tuliskan semua simpul tetangga termasuk dari iterasi sebelumnya yang belum diperiksa]

Dst... (Catatan: pencarian dihentikan ketika simpul yang diperiksa = F)

Jalur hasil pencarian dengan BFS =

Jarak jalur hasil pencarian dengan BFS =

DFS

Iterasi 1:

Simpul Ekspan (Simpul yang diperiksa) = A

Simpul Hidup = B[A] C[A] D[A] (catatan: C[AB] artinya simpul C dari jalur AB)

Iterasi 2:

Simpul Ekspan (Simpul yang diperiksa) = [tuliskan simpul yang diperiksa berikutnya]

Simpul Hidup = [tuliskan semua simpul tetangga termasuk dari iterasi sebelumnya yang belum diperiksa]

Dst... (Catatan: pencarian dihentikan ketika simpul yang diperiksa = F)

Jalur hasil pencarian dengan DFS =

Jarak jalur hasil pencarian dengan DFS =

Soal Pilihan

10. Algoritma greedy untuk kompresi yang pertama kali dikemukakan oleh Huffmann mempunyai kompleksitas sebagai berikut :

Tandai satu oval saja.

- $O(n)$
- $O(\log n)$
- $O(n \log n)$
- $O(n*n)$
- Semua jawaban salah

11. Manakah diantara pernyataan di bawah ini yang paling benar berdasarkan time complexity nya :

Tandai satu oval saja.

- Algoritma Prim lebih baik daripada algoritma Kruskal
- Algoritma Kruskal lebih baik dari Algoritma Prim
- Dalam kasus graph yang fully connected Algoritma Kruskal lebih baik daripada algoritma Prim
- Dalam kasus graph yang fully connected Algoritma Prim lebih baik daripada algoritma Kruskal
- Semua jawaban salah

12. Pilihlah mana pernyataan di bawah ini yang salah :

Tandai satu oval saja.

- Algoritma Greedy dapat mencari solusi paling optimal dalam persoalan activity selection problem
- Algoritma Greedy dapat mencari solusi paling optimal dalam persoalan shortest-path problem
- Algoritma Greedy dapat mencari solusi paling optimal dalam persoalan fractional knapsack problem
- Algoritma Greedy dapat mencari solusi paling optimal dalam persoalan integer 0/1 knapsack problem
- Semua jawaban salah

13. Kompleksitas waktu sebuah algoritma divide and conquer berbentuk relasi rekurens $T(n) = T(2n/3) + 2$. Maka, notasi Big-O kompleksitas waktunya adalah

Tandai satu oval saja.

- $O(n)$
- $O(\log n)$
- $O(n \log n)$
- $O(n^2)$
- $O(n^2 \log n)$
- Tidak ada jawaban yang benar

14. Sebuah persoalan fractional knapsack terdiri dari 4 buah objek dengan bobot (weight) dan profit dinyatakan dalam tabel berikut. Kapasitas knapsack adalah 10. Solusi optimal persoalan fractional knapsack tersebut adalah memiliki keuntungan:

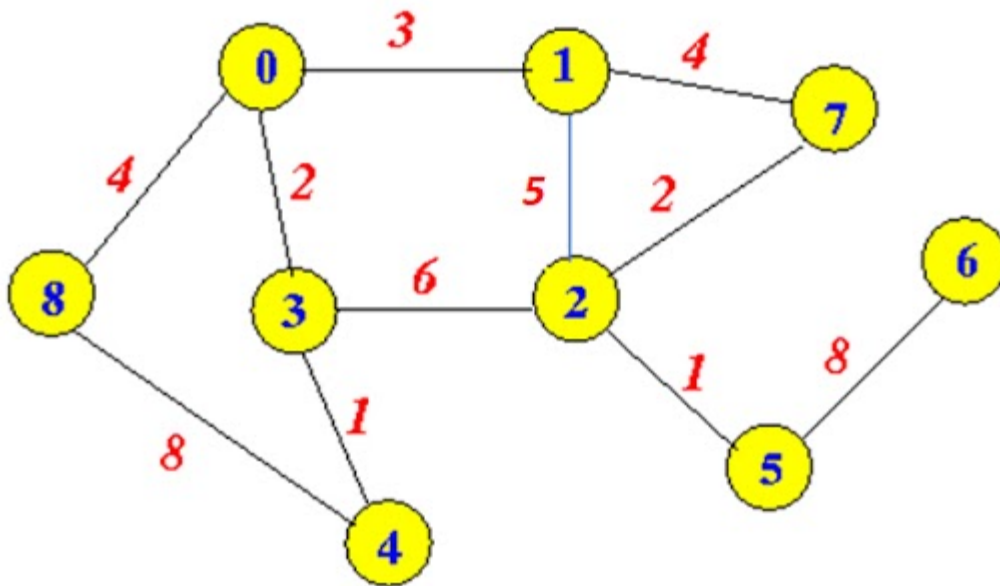
Items	1	2	3	4
Weights	5	4	6	3
Profit	10	40	30	50

Capacity : 10

Tandai satu oval saja.

- 95
- 100
- 105
- 110
- 120
- Tidak ada jawaban yang memenuhi

15. Diberikan sebuah graf berbobot sebagai berikut. Misalkan initial state adalah dari simpul 0 dan goal state adalah simpul 6. Jika rute dari 0 ke 6 dicari secara aturan BFS, maka urutan simpul-simpul yang dilalui dan total bobotnya adalah: (Catatan: urutan simpul yang dibangkitkan sesuai dengan nomor urut dari kecil ke besar)



Tandai satu oval saja.

- 0, 1, 7, 2, 5, 6; bobot = 18
- 0, 3, 2, 5, 6, bobot = 17
- 0, 8, 4, 3, 2, 5, 6, bobot = 28
- 0, 1, 2, 5, 6, bobot = 17
- Tidak ada jawaban yang benar

16. QuickSort adalah metode pengurutan dengan divide and conquer dimana sebuah larik A yang akan diurutkan (dalam tahapan divide) dipartisi menjadi dua larik A1 dan A2 sehingga semua elemen di $A1 \leq$ semua elemen di A2 jika akan diurut menaik, atau sebaliknya jika akan diurut menurun. Misalkan A akan diurut menurun (descending order) dan berisi deretan elemen 5, 8, 2, 1, 4, 6, 9, 3, 7, maka partisi A1 dan A2 yang akan dihasilkan apabila pivot untuk proses partisi diambil dari elemen pertama adalah:

Tandai satu oval saja.

- A1 = 5, 8, 7, 9, 6 dan A2 = 4, 1, 3, 2
- A1 = 5, 3, 2, 1, 4 dan A2 = 6, 9, 8, 7
- A1 = 8, 7, 9, 6 dan A2 = 5, 4, 1, 3, 2
- A1 = 3, 2, 1, 4 dan A2 = 5, 6, 9, 8, 7
- Tidak ada jawaban yang benar

17. Misalkan pencarian nilai minimum dan maksimum dalam suatu larik dilakukan dengan menerapkan Divide dan Conquer, yaitu dengan cara membagi secara rekursif menjadi dua sublarik sampai ke ukuran yang dapat ditentukan langsung nilai minimum dan maksimumnya (saat sublarik berukuran 1 atau 2). Jika sebuah larik mengandung deretan elemen berikut: 11, 21, 6, 45, 3, 19, 64, 10, 50 maka jumlah perbandingan nilai elemen yang diperlukan untuk mendapatkan elemen dengan nilai minimum dan maksimum adalah:

Tandai satu oval saja.

- 10
- 11
- 12
- 13
- 14

18. Algoritma MergeSort menerapkan Divide and Conquer dengan membagi larik menjadi dua sublarik secara rekursif dan melakukan penggabungan dengan fungsi Merge dari dua sublarik yang sudah terurut. Jika dua sublarik tersebut adalah $A1 = 4, 12, 14, 25, 40$ dan $A2 = 2, 11, 15, 22, 24$ maka jumlah perbandingan nilai elemen saat mengeksekusi fungsi Merge tersebut adalah:

Tandai satu oval saja.

9

6

7

10

8

19. Terdapat persoalan n-ratu, di mana n ratu harus diletakkan pada papan catur berukuran n^2 , dan tidak ada ratu yang berada pada baris yang sama, kolom yang sama, dan diagonal yang sama. Berapa kemungkinan banyak posisi yang harus diperiksa dengan pendekatan Exhaustive Search ketika saat meletakkan ratu tidak boleh ada ratu yang berada pada baris dan kolom yang sama?

Tandai satu oval saja.

a. Kombinasi: $C(n^2, n)$

b. n^n

c. $n!$

d. Permutasi: $P(n^2, n)$

e. Tidak ada jawaban yang benar

20. Jika terdapat sebuah kalimat KULIAH_STRATEGI_ALGORITMA_SANGAT_MENANTANG dan ingin dicari apakah pola UJIAN muncul pada kalimat tersebut, maka banyaknya perbandingan yang dilakukan dengan pendekatan Brute Force adalah:

Tandai satu oval saja.

- a. 36
- b. 37
- c. 38
- d. 39
- e. 40
- f. 41
- g. 42
- h. Tidak ada yang benar.

21. Dari pernyataan berikut, tentukan apakah pernyataan tersebut benar atau salah.

Tandai satu oval saja per baris.

	Benar	Salah
a. Pendekatan Brute Force belum tentu mendapatkan solusi dari suatu persoalan, walaupun solusinya ada.	<input type="radio"/>	<input checked="" type="radio"/>
b. Jika Topological Sort dapat diterapkan pada suatu graf berarah, maka graf tersebut mungkin membentuk sirkuit.	<input type="radio"/>	<input checked="" type="radio"/>
c. Persoalan apakah sebuah elemen terdapat pada suatu larik, paling mangkus diselesaikan dengan pendekatan Sequential Search yang memiliki kompleksitas $O(n)$.	<input type="radio"/>	<input checked="" type="radio"/>

Konten ini tidak dibuat atau didukung oleh Google.

Google Formulir