

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

### **Brute Force + Greedy**

- Misalkan terdapat sebuah larik  $a[1..n]$  dengan  $n$  elemen bilangan bulat. Kita ingin menghitung  $F = \sum_{i=1}^n i * a[i]$  sedemikian sehingga  $F$  bernilai maksimum.
  - Jika diselesaikan secara *brute force/exhaustive search*, bagaimana caranya, dan berapa perkiraan kompleksitas waktunya (dalam notasi Big-Oh)? (5)
  - Jika diselesaikan secara *greedy*, bagaimana caranya, dan berapa perkiraan kompleksitas waktunya? Contohkan jawaban anda untuk larik  $a = [3, 5, 6, 1]$ . (10)
- Diberikan  $n$  buah job. Setiap job  $i$  memiliki  $(s_i, f_i)$ ,  $s_i$  = waktu mulai,  $f_i$  = waktu selesai,  $s_i < f_i$ . Tersedia banyak mesin untuk mengerjakan semua job tersebut. Satu mesin dapat mengerjakan job-job secara sekuensial asalkan waktunya tidak bentrok (beririsan). Setiap job dikerjakan dari waktu mulai hingga waktu selesainya. Berapa *minimal* jumlah mesin yang dibutuhkan untuk mengerjakan **semua** job tersebut? Sebagai contoh:  $n = 4$ ,  $(s_i, f_i) = [(4,8), (1,3), (3,4), (4,7)]$ , dibutuhkan minimal dua mesin, yaitu mesin 1:  $[(1,3), (4, 7)]$ , mesin 2 :  $[(3, 4), (4, 8)]$ . Jika persoalan ini diselesaikan dengan algoritma *greedy*, jelaskan bagaimana strateginya, dan tentukan berapa kompleksitas waktu asimptotiknya. Jelaskan jawaban anda untuk contoh berikut:  $n = 8$ ,  $(s_i, f_i) = [(4,7), (2,5), (1,4), (3,7), (7,8), (1,3), (6,9), (5, 8)]$ , lalu tentukan berapa jumlah mesin yang dibutuhkan. (15)

### **Divide and Conquer**

- Jelaskanlah strategi umum Brute Force dan Divide and Conquer Karatsuba pada persoalan perkalian dua bilangan besar X dan Y, dan lengkapi kompleksitas waktunya (dalam notasi Big-O). Gambarkan pohon pemanggilan rekursif D&C dari perkalian 2412 dan 3231. (15)
- MergeSort merupakan salah satu teknik pengurutan dengan Divide and Conquer yang memiliki kompleksitas algoritma  $O(n \log_2 n)$ . Jika diberikan suatu array sembarang berukuran  $n=2^k$ , jelaskanlah berapa kali pemanggilan rekursif MergeSort dan Merge dilakukan dalam pengurutan array tersebut. Pemanggilan pertama MergeSort(A,1,n) masuk dalam perhitungan. (5)
  - Lakukanlah proses divide atau partisi pada QuickSort untuk array karakter 'MARET' dengan menggunakan pivot elemen tengah sehingga array terurut menaik. Indeks array dimulai dari 1. Pada setiap langkah perjas posisi yang ditukar, dan posisi pada tabel di mana partisi dilakukan. (7.5)
  - Diberikan kumpulan titik  $\{[1, 1], [2, 2], [4, 4], [0, 0], [1, 2], [3, 1], [3, 3]\}$ , lakukanlah proses QuickHull yang menggunakan strategi Divide and Conquer untuk mendapatkan kumpulan titik yang membentuk convex hull. Tidak perlu melakukan perhitungan jarak antara dua titik, boleh menggunakan estimasi untuk prosesnya. (7.5)

**Decrease and Conquer + DFS-BFS**

5. Lengkapi tabel berikut ini sesuai petunjuk di tiap soal.

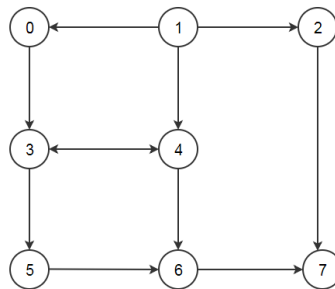
(a) Isikan perbandingan antara ketiga teknik dalam tabel berikut ini. (9)

Aspek	Binary Search	Interpolation Search	Pencarian Median (Selection Problem dengan $k = \lfloor n/2 \rfloor$ )
Decrease by :			
Larik harus terurut (Ya/ Tidak)			
Kompleksitas Algoritma (Big O):			

(b) Terdapat sebuah larik unik A sebagai berikut:  $A = [3, 14, 27, 31, 39, 42, 55, 70, 74, 81, 85, 93, 98]$ . Carilah indeks di mana nilai  $K = 85$  berada (indeks larik dimulai dari indeks 1), dengan pendekatan Binary Search dan Interpolation Search. Jika tidak ditemukan bilangan tersebut pada larik, pencarian menghasilkan -1. Tuliskan proses pencarian dengan melengkapi tabel berikut ini (penentuan nilai mid dan iterasi untuk tiap jenis pencarian). (11)

	Binary Search			Interpolation Search		
Formula pencarian indeks mid:	mid = ...			mid = ...		
Iterasi	Indeks awal	Indeks akhir	Indeks mid	Indeks awal	Indeks akhir	Indeks mid
1	1	13	...	1	13	...
2	...	...	...	...	...	...
Dst...	...	...	...	...	...	...
Indeks Akhir hasil pencarian:	...			...		
Jumlah Iterasi	...			...		

6. Terdapat sebuah gambar graf berarah sebagai berikut.



(a) Jika semua simpul harus dikunjungi, tuliskan urutan simpul yang dikunjungi dimulai dari simpul 1, dengan teknik: (i) BFS; (ii) DFS. Catatan: prioritas simpul dengan aturan 'ascending'. (5)

(b) Bentuklah pohon pencarian jalur dari simpul 1 ke simpul 7 dengan pendekatan DFS dan BFS dengan melengkapi tabel berikut ini. Simpul yang belum pernah di ekspan akan tetap dibangkitkan dan masuk dalam simpul hidup. Simpul yang sudah pernah di ekspan (dari jalur manapun) tidak perlu dibangkitkan. (10)

BFS		DFS	
Simpul Ekspan	Simpul Hidup	Simpul Ekspan	Simpul Hidup
1	0 <sub>1</sub> , 2 <sub>1</sub> , 4 <sub>1</sub>	1	0 <sub>1</sub> , 2 <sub>1</sub> , 4 <sub>1</sub>
...	...	...	...
Dst...	...	Dst...	...
Hasil pencarian rute dari simpul 1 ke simpul 7: ...		Hasil pencarian rute dari simpul 1 ke simpul 7: ...	