

Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!

Bagian A (Soal dari MLK)



Pada persoalan *muddy city*, terdapat suatu kota tanpa jalan aspal. Jika hujan badai, jalanan menjadi sangat berlumpur, mobil terjebak di lumpur dan boots orang menjadi kotor. Walikota memutuskan untuk memasang *paving block* lebar, tetapi hanya ingin menghabiskan biaya seminimal mungkin. Terdapat dua kondisi: (1) *Paving block* cukup dipasang sehingga penduduk dapat datang

dari rumah mereka ke rumah lain melalui jalan ber-*paving block*, mungkin saja harus melalui rumah lainnya; (2) Jumlah *paving block* seminimal mungkin agar biayanya juga minimal. Bantulah Pak Walikota dengan memberikan solusi jalur *paving block* yang akan dipasang. (Sumber gambar: <http://statkclee.github.io/website-csunplugged>)

- a. (Nilai 5) Representasikanlah gambar kota di atas menjadi sebuah graf, dengan simpul merepresentasikan rumah, dan bobot sisi merepresentasikan jumlah *paving block* yang dibutuhkan. Jembatan dihitung sebagai satu *paving block*.
- b. (Nilai 15) Jika diselesaikan dengan *exhaustive search*, jelaskan strateginya seperti apa (tidak perlu pseudo-code), lalu tentukanlah kompleksitas algoritmanya dalam notasi big O.
- (Nilai 15) Jika diselesaikan dengan *greedy*, rancanglah strategi *greedy* terbaiknya, dan tentukanlah kompleksitas algoritmanya dalam notasi big O.

Bagian B (Soal dari NUM)

- Aplikasikan algoritma **Merge Sort** dan **Quick Sort** pada kumpulan huruf pada tabel berikut ini menjadi terurut abjad.

E	X	A	M	P	L	E
---	---	---	---	---	---	---

Gambarkan pohon pemanggilan rekursif untuk **masing-masing** algoritma tersebut. Untuk **Quick Sort**, pivot yang digunakan adalah elemen pertama tabel. Pada setiap langkah perjas posisi yang ditukar, dan posisi pada tabel di mana partisi dilakukan.

Untuk **setiap** algoritma, tentukan berapa kali pemanggilan rekursif dilakukan dalam rangka mendapatkan kumpulan huruf yang sudah terurut abjad. (Nilai: 15)

4. Terdapat sebuah larik A yang berisi bilangan bulat positif dan negatif sebagai berikut. (Nilai: 20)

{2, -4, 1, 9, -6, 7, -3}

- Rancanglah pendekatan *Divide and Conquer* untuk menemukan bagian upa-larik yang sekuensnya memiliki jumlah maksimum (yang berlaku untuk semua larik dan tidak hanya pada contoh larik A di atas). Misal i adalah indeks awal dan j adalah indeks akhir suatu sekuens upa-larik A , maka dicari $S_{ij} = \sum_{k=i}^j A[k]$ bernilai maksimal. Contoh untuk larik A yang indeksnya dimulai dari 1, maka upa-larik yang menghasilkan sekuens nilai maksimal adalah $i=3$ dan $j=6$, yang jumlah elemennya adalah 11. Rancangan tidak perlu sampai *pseudo code*, cukup penjelasan langkah-langkah.
- Tentukan waktu yang diperlukan dari pendekatan yang anda usulkan dalam notasi $T(n)$, dengan menentukan basis dan rekurensya.
- Tentukan kompleksitas dari pendekatan yang anda usulkan dalam notasi Big O, dengan memanfaatkan Teorema Master (perjelas nilai a , b , dan d sesuai jawaban anda di butir (b)).

Bagian C (Soal dari RN)

5. (*Selection problem*) Diberikan larik (*array*) sebagai berikut:

13, 9, 18, 6, 8, 11, 15, 7, 12

Perlihatkan proses mencari elemen terbesar ke-5 dengan algoritma *decrease and conquer* dan memanfaatkan algoritma partisi dari algoritma *Quicksort* varian kedua. *Pivot* yang diambil selalu elemen pertama larik. (Nilai: 12)

6. (*Block World Problem*) Terdapat beberapa buah balok berbentuk kubus yang ditempatkan di atas meja atau di atas balok yang lain sehingga membentuk sebuah konfigurasi. Sebuah robot yang memiliki lengan bercapit harus memindahkan balok-balok kubus tersebut sehingga membentuk konfigurasi lain dengan jumlah perpindahan yang minimum. Persyaratannya adalah hanya boleh memindahkan satu balok setiap kali ke atas balok lain atau ke atas meja. Gambarkan pohon ruang status pencarian solusi secara BFS dan DFS untuk *initial state* dan *goal state* di bawah ini. Setiap status digambarkan sebagai tumpukan balok kubus setelah pemindahan satu balok. Beri nomor setiap status sesuai aturan BFS dan DFS. Hitung berapa banyak status yang dibangkitkan sampai ditemukan *goal state*. (Nilai: 18)

