Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Ujian Tengah Semester IF2211 Strategi Algoritma

Hari Jumat, Tanggal 13 Maret 2015

Waktu: 120 menit

Dosen: Rinaldi Munir & Nur Ulfa Maulidevi

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

**Bagian A (Soal dari Dr. Rinaldi Munir)**

1. (***Brute Force, Divide and Conquer***) Seorang ahli biologi sedang meneliti *n* sampel DNA. Menemukan kode DNA eksak untuk tiap sampel memakan waktu yang lama. Namun, semua ahli biologi ingin mengetahui apakah terdapat paling sedikit setengah dari sampel berasal dari hewan yang sama. Dia ingin memperoleh metode yang dengan cepat menentukan apakah sembarang pasangan sampel DNA berasal dari hewan yang sama. Dengan mengasumikan *n* adalah perpangkatan dari dua, maka:
2. Bagaimana algoritma *brute force* untuk menentukan apakah paling sedikit setengah dari *n* sampel DNA tersebut berasal dari hewan yang sama. Deskripsikan algoritmanya (bukan *pseudo-code*) dan perkirakan kompleksitas algoritmanya dalam notasi O-besar.
3. Rancanglah algoritma *divide and conquer* yang menentukan apakah paling sedikit dari *n* sampel DNA berasal dari hewan yang sama. Deskripsikan algoritmanya masing-masing untuk bagian basis dan bagian rekurens (bukan *pseudo-code*), hitung kompleksitas waktunya dalam *T*(*n*) yang berbentuk rekursif , lalu perkirakan kompleksitas algoritmanya dalam notasi O-besar (menggunakan Teorema Master).

**(Nilai = 10 + 10)**

1. **(Teorema Master)** Gunakan teorema Master untuk menentukan notasi asymptotic untuk *T*(*n*) berikut:
2. *T*(n) = 8*T*(*n*/2) + *n*2
3. *T*(*n*) = 2*T*(*n*/2) + *n*2
4. *T*(*n*) = 4*T*(*n*/2) + *n*2
5. (Petunjuk: misalkan *n* =2*m*, lalu definisikan S(*m*) = *T*(2*m*). Jelaslah bahwa solusi *S*(*m*) juga solusi *T*(*n*))

**(Nilai = 3 + 3 + 3 + 6)**

1. **(*Brute Force, Divide and Conquer, Decrease and Conquer)*** Menghitungperpangkatan *an*, *n* adalah bilangan bulat ≥ 0. Misalkan *T*(*n*) adalah jumlah operasi perkalian yang terjadi.
2. Rumuskan cara perhitungan dengan algoritma *brute force*, lalu hitung *T*(*n*).
3. Rumuskan cara perhitungan dengan algoritma *divide and conquer*, lalu hitung *T*(*n*).
4. Rumuskan cara perhitungan dengan algoritma *decrease and conquer*, lalu hitung *T*(*n*).

**(Nilai = 5 + 5 + 5)**

**Bagian B (Soal dari Dr. Nur Ulfa Maulidevi)**

Untuk soal nomor 4 dan 5, perhatikan gambar graf tak berarah berikut ini. Setiap simpul merepresentasikan router pada suatu jaringan komputer. Simpul yang menghubungkan dua buah router merepresentasikan biaya (*cost*) jika melewati jalur tersebut.



1. **(Algoritma *Greedy*)**
	1. Tentukan matriks ketetanggaan dari graf tak berarah tersebut, dengan isi tiap sel matrik adalah bobot yang bersesuaian.
	2. Gunakan algoritma Djikstra untuk mencari jalur terpendek dari simpul **u** ke simpul yang lain. Dalam proses penulisan solusi, gunakan tabel sebagai berikut.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lelaran ke- | Simpul yang dipilih | Lintasan | Simpul sudah dikunjungi(1= sudah, 0 = belum) | Jarak dari simpul u |
| u | v | w | x | y | z | u | v | w | x | y | z |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Catatan: Pada kolom Jarak dari simpul u, tentukan lintasan yang digunakan untuk mendapatkan jarak tersebut.

* 1. Berdasarkan jawaban anda pada butir (b), tentukan jalur minimal yang menghubungkan simpul u dengan simpul z.
	2. Tentukan kompleksitas waktu asimptot untuk mencari jalur terpendek dengan algoritma ini.

 **(Nilai = 30)**

1. **(Soal BFS dan DFS)**
	1. Dengan menggunakan pendekatan **DFS**, carilah jalur dari simpul **u** ke simpul **z**. Gambarkan pohon proses penelusuran graf yang dilakukan. Asumsi: urutan simpul sesuai dengan urutan abjad.
	2. Dengan menggunakan pendekatan **BFS**, carilah jalur dari simpul **u** ke simpul **z**. Gambarkan pohon proses penelusuran graf yang dilakukan. Asumsi: urutan simpul sesuai dengan urutan abjad.
	3. Jika b adalah banyaknya anak maksimum yang bisa di ekspansi oleh suatu node, d adalah kedalaman minimum untuk mencapai solusi, dan m adalah kedalaman maksimum yang mungkin dilakukan saat proses pencarian solusi, bagaimanakah kompleksitas waktu dan kompleksitas ruang untuk masing-masing algoritma DFS dan BFS?
	4. Berdasarkan jawaban anda di 5(a) dan 5(b), algoritma manakah yang memberikan solusi lebih baik jika dilihat dari:
		1. Banyaknya perpindahan router (banyaknya *hops*) yang dilakukan untuk mencapai simpul z dari simpul u (jelaskan dengan singkat).
		2. Biaya yang dibutuhkan untuk mencapai simpul z dari simpul u (jelaskan dengan singkat).
		3. Bandingkan dengan jalur terpendek yang dihasilkan oleh Algoritma Djikstra.

 **(Nilai = 20)**