

---

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama  
Sidang Akademik 2004/2005

Oktober 2004

**CPT201 – Reka Bentuk & Analisis Algoritma**

Masa : 2 jam

---

**ARAHAN KEPADA CALON:**

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **LIMA** soalan di dalam **LIMA** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
  - Jawab mana-mana **EMPAT** soalan.
-

1. (a) Susun senarai fungsi berikut mengikut tatatanda O besar (kadar perkembangan terkecil di sebelah kiri), dan kumpulkan (umpamanya dengan membulatkan bersama) fungsi-fungsi yang merupakan O besar bagi fungsi yang lain.

$$6N \log N \quad 4N^{3/2} \quad 2^N \quad N \log_4 N \quad 5N \quad 4^N \quad 4^{\log N} \quad N^3$$

(20/100)

- (b) Beri satu algoritma  $O(mn)$  untuk menentukan sama ada set A merupakan subset kepada set B. Buktikan bahawa algoritma anda dijalankan dalam masa tersebut.

(30/1000)

- (c) Anda dikehendaki mengisih satu tatasusunan yang besar yang menyebabkan penggunaan algoritma  $O(N^2)$  dan data tambahan (umpamanya tindakan) mungkin tidak memuaskan. Walau bagaimanapun anda diberitahu bahawa tatasusunan yang ingin diisih selalunya hampir terisih keadaannya. Adakah sesuai jika isihan cantum digunakan untuk mengisih tatasusunan tersebut? Apakah algoritma pengisihan yang paling sesuai untuk menjalankan tugas berkenaan? Jelaskan jawapan-jawapan anda.

(30/100)

- (d) Kenal pasti dan beri satu contoh set data bagi kes terbaik dan kes terburuk bagi algoritma isihan cantum.

(20/100)

2. (a) (i) Apakah yang anda harus lakukan terhadap algoritma isihan pepohon jika anda perlu mengisih satu senarai ke dalam tertib menurun dan bukannya tertib menaik?

- (ii) Surih algoritma isihan pepohon yang mengisih tatasusunan 40 30 50 20 60 70 ke dalam tertib menurun seperti yang anda cadangkan dalam 2(a)(i) di atas.

(30/100)

- (b) (i) Senaraikan pengendalian-pengendalian yang menakrif ADT jadual bersama-sama dengan huraian ringkas bagi setiap pengendalian.

- (ii) Pengendalian ADT jadual yang manakah boleh digunakan untuk mengosongkan sebuah jadual yang sedia ada bersasaskan tatasusunan? Huraikan bagaimana anda menggunakan pengendalian berkenaan untuk menjalankan tugas tersebut.

(30/100)

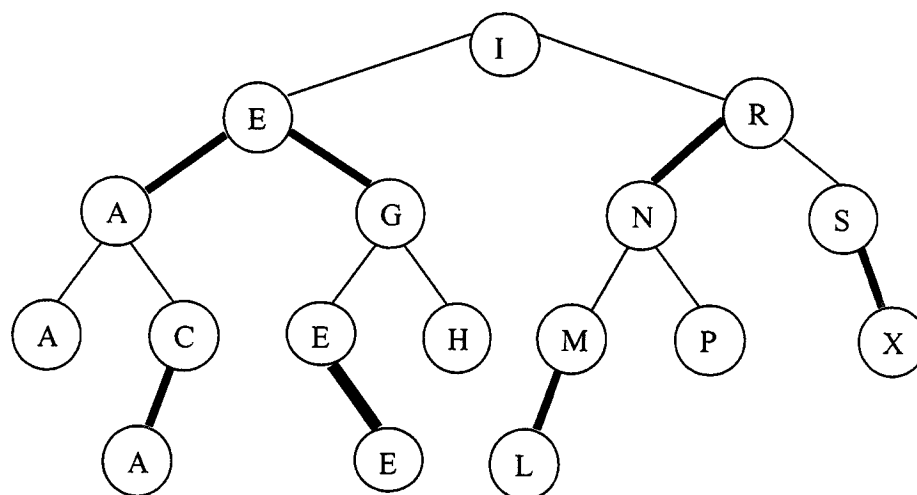
(c) Terangkan apa yang akan terjadi dalam sebutan pergerakan dan perbandingan, di dalam HeapInsert (penyisipan ke dalam timbunan seperti yang diberikan dalam kuliah) jika data asal

- (i) dalam tertib terisih.
- (ii) terdiri daripada kunci-kunci yang sama nilai.
- (iii) adalah tersedia sebuah timbunan.
- (iv) dalam tertib terisih terbalik.

Ilustrasikan jawapan anda dengan menggunakan set data berikut: masing-masing 1 2 3 4 5 bagi 2(c)(i), 2 2 2 2 2 bagi 2(c)(ii), 5 4 3 1 2 bagi 2(c)(iii), dan 5 4 3 2 1 bagi 2(c)(iv).

(40/100)

3. (a) (i) Huraikan konsep pepohon merah hitam dan pepohon 2-3-4.
- (ii) Mengapakah kita menggunakan perwakilan merah hitam untuk mewakili pepohon 2-3-4?
- (iii) Tulis algoritma (pseudokod) penyusunan tertib sisipan yang akan melawat nod-nod pepohon merah hitam dan pada masa sama mencetak jenis pautan iaitu pautan merah atau hitam, yang dilaluinya.
- (iv) Apakah yang akan dicetak tentang jenis pautan yang ada jika algoritma dalam 3(a)(iii) di atas dijalankan ke atas pepohon merah-hitam di bawah? (Catatan: Pautan merah – garis tebal, pautan hitam – garis biasa)



- (v) Apakah pepohon 2-3-4 yang mewakili pepohon merah hitam yang diberikan dalam 3(b)(iv) di atas?

(60/100)

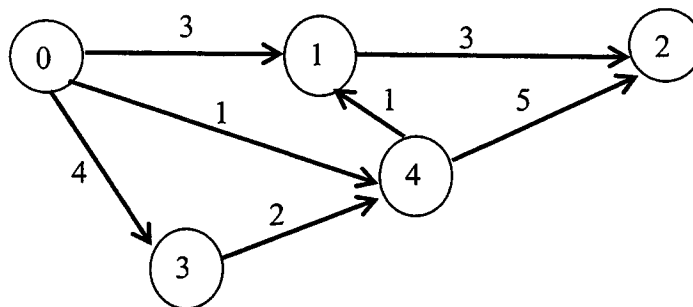
- (b) Berikut ialah jadual cincangan bagi 10 integer yang dicipta menggunakan fungsi cincangan  $h_1(\text{kunci}) = \text{kunci} \% 10$ . Perlanggaran dileraikan dengan menggunakan pencarian linear (linear probing).

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
10	430	21	23	54	235	51	17	37	28

- Adakah fungsi cincangan tersebut fungsi cincangan yang baik? Jelaskan jawapan anda.
- Beri satu jujukan penyisipan yang mungkin untuk menyisip integer-integer berkenaan ke dalam jadual cincangan tersebut yang asalnya kosong.
- Bagi setiap integer, hitung bilangan perbandingan yang diperlukan untuk mencari integer berkenaan. Dengan menggunakan keputusan berkenaan, hitung purata bilangan perbandingan bagi sesuatu gelintaran di dalam jadual cincangan ini.

(40/100)

4. (a) (i) Mengapakah perwakilan matriks kesebelahan sesuai bagi algoritma lintasan terpendek Dijkstra dan bukannya perwakilan senarai kesebelahan?
- (ii) Lakukan algoritma lintasan terpendek Dijkstra ke atas graf berikut dengan menyuruhnya menggunakan format berjadual seperti yang diberikan dalam kuliah.



(45/100)

- (b) Kita menyatakan bahawa pepohon rentang minimum akan sentiasa mengandungi  $N-1$  tepi bagi graf  $N$  bucu. Beri hujah logik untuk membuktikan kenyataan ini. (Petunjuk: Gunakan bukti dengan penyangkalan (contradiction).)

(15/100)

- (c) (i) Huraikan fungsi ReadBlock dan WriteBlock (seperti yang diberikan dalam kuliah) dan nyatakan alasan mengapa kita memerlukan fungsi-fungsi ini bagi kaedah luaran.

- (ii) Tulis fungsi (pseudokod) yang akan membaca blok-blok sebuah fail luaran secara berjujukan dan memproses (melawat) rekod-rekod secara berjujukan di dalam setiap blok dan menulis semula rekod yang dikemas kini ke dalam fail.

(40/100)

5. (a) (i) Huraikan kaedah penyuaian pertama dan kaedah penyuaian terbaik.
- (ii) Yang manakah antara kaedah-kaedah ini merupakan kaedah yang lebih baik?

(25/100)

- (b) Huraikan dengan menggunakan perkataan anda sendiri algoritma DisposePtr (seperti yang diberikan dalam kuliah) dan menggunakan ilustrasi bergrafik jika bersesuaian.

(30/100)

- (c) Algoritma berikut mencari satu subrentetan di dalam satu rentetan yang diberikan. Algoritma berkenaan mengembalikan kedudukan dalam sumber, kejadian *pertama* subrentetan *target*, dan mengembalikan -1 jika tidak berjaya.

```
int Find (const String& source, const String& target)
{
    if (source or target is empty)
        return -1; // no match possible
    int current = 0; // possible location in this
    while (complete match not found
           && any characters left in source)
        if (current character in source != first target character)
            current++;
        else {do // found a partial match
                Step through source and target together
                while (chars left to compare && still have a match);
                if (no more target characters to inspect)
                    return current; // found a full match.
                else current++; // keep looking
            }
    return -1; // no match found
}
```

Ubahsuai algoritma ini supaya semua kejadian berkenaan dilokasikan dan semua kedudukan tempat kejadian berkenaan dijumpai diletakkan di dalam sebuah baris gilir. Gunakan ADT baris gilir.

(25/100)

- (d) Hasilkan kod Huffman bagi abjad a, b, c, d dan huruf-huruf ini mempunyai masing-masing kekerapan: 0.3, 0.25, 0.20, 0.25.

(20/100)