

---

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama  
Sidang Akademik 2002/2003

September 2002

**CPT201/CTP201 – Reka Bentuk & Analisis Algoritma**

Masa : 3 jam

---

**ARAHAN KEPADA CALON:**

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

1. (a) (i) Susun ungkapan-ungkapan berikut mengikut kadar perkembangan daripada paling cepat kepada paling perlahan.

$$n \quad \log_2 n \quad n^2 \quad n \log_2 n \quad n^3 \quad 2^n \quad 1$$

- (ii) Jelaskan mengapa konsep kadar perkembangan sangat penting. (15/100)

- (b) Pertimbangkan masalah untuk mencari nilai yang paling banyak wujud di dalam tatasusunan terisih.

- (i) Huraikan satu penyelesaian yang mungkin kepada algoritma berkenaan, dan tulis pseudokod untuk algoritma tersebut.

- (ii) Apakah kekompleksan/kecekapan algoritma ini dalam tatatanda O? (25/100)

- (c) Hampir kesemua algoritma yang anda telah pelajari beroperasi ke atas satu tatasusunan. Apakah kesannya ke atas algoritma berikut jika algoritma-algoritma berkenaan beroperasi ke atas satu senarai berpaut? Bagi setiap algoritma berkenaan, huraikan satu penyelesaian yang mungkin (jangan tulis sebarang kod), masalah-masalah yang dihadapi (jika ada) dan sama ada pelaksanaan senarai berpaut akan memberi kesan kepada kekompleksan/ kecekapan algoritma berkenaan.

- (i) Isihan sisip.

- (ii) Isihan cantum.

- (iii) Isihan cepat. (30/100)

- (d) (i) Fungsi `quicksort` (isihan cepat – seperti yang diberikan dalam kuliah) menggunakan fungsi `choosePivot` (pilih pangsi – seperti yang diberikan dalam kuliah) untuk memilih satu pangsi dan meletakkannya ke dalam lokasi pertama tatasusunan. Huraikan dua cara yang anda tahu untuk melaksanakan `choosePivot`.

- (ii) Lakukan isihan cepat ke atas integer berikut menggunakan setiap daripada dua cara yang anda berikan dalam (i) di atas.

26 38 10 35 26 7

(30/100)

2. (a) Idea asas tentang isihan pepohon adalah seperti yang berikut (seperti yang diberikan dalam kuliah).

```
Treesort (inout anArray: ArrayType, in n: integer)
// Mengisih n integer di dalam tatasusunan anArray ke dalam
// tertib menaik.
```

Sisip unsur anArray ke dalam pepohon gelintaran perduaan  
bTree.

Susur bTree mengikut tertib sisipan. Sementara anda melawat nod-  
nod bTree's, salin item data berkenaan ke dalam lokasi  
berturutan.

- (i) Dengan menggunakan perkataan anda sendiri, huraikan dengan terperinci algoritma di atas.
- (ii) Bagaimanakah anda mengubahsuai algoritma di atas supaya kita dapat mengisih satu tatususunan ke dalam tertib menurun dan bukannya tertib menaik?

(25/100)

- (b) Pelaksanaan tak linear, pepohon gelintaran perduaan untuk ADT jadual merupakan satu pilihan yang baik secara umum jika dibandingkan dengan dua kaedah pelaksanaan linear. Berikan kebaikan dan keburukan pelaksanaan pepohon gelintaran perduaan untuk ADT jadual, dengan memberikan perhatian terhadap kebaikannya ke atas dua kaedah pelaksanaan linear.

(25/100)

- (c) Dalam takrifan kita bagi timbunan, akar mengandungi item dengan kunci gelintaran terbesar. Timbunan seperti ini digelar makstimbunan. Satu mintimbunan pula, meletakkan item dengan kunci gelintaran terkecil di akarnya.

- (i) Tulis semula takrifan timbunan berasaskan mintimbunan.
- (ii) Di manakah unsur terbesar mungkin berada di dalam mintimbunan?
- (iii) Tunjukkan mintimbunan yang terhasil daripada penyisipan integer berikut ke dalam timbunan dalam tertib yang diberikan.

10 5 12 8

- (iv) Apakah yang terjadi jika fungsi `heapsort` (isihan timbun – seperti yang diberikan dalam kuliah) untuk makstimbunan, dikenakan ke atas satu mintimbunan (tanpa sebarang pengubahsuaian terhadap fungsi berkenaan)? Jelaskan.
- (v) Dengan mengandaikan fungsi `heapRebuild` (bina semula timbunan – seperti yang dirikan dalam kuliah) telah diubahsuai untuk mintimbunan, lakukan isihan timbun (ke dalam tertib menurun) ke atas mintimbunan yang diperolehi dalam (ii) di atas.

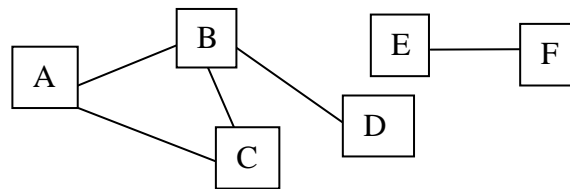
(50/100)

3. (a) Beri kekompleksan/kecekapan penggelintaran bagi setiap struktur gelintaran berikut.
- (i) Pepohon 2-3.
  - (ii) Pepohon 2-3-4.
  - (iii) Pepohon merah-hitam.
  - (iv) Pepohon AVL.
- (20/100)
- (b) Jika aplikasi ADT jadual hanya melibatkan capaian kembali seperti tesaurus – pepohon manakah dari (a) di atas akan menyediakan pelaksanaan jadual yang paling cekap? Justifikasikan jawapan anda.
- (15/100)
- (c) Huraikan strategi asas kaedah AVL dalam pengimbangan pepohon gelintaran perduaan. Huraian anda mestilah merangkumi pelbagai jenis putaran yang terlibat serta beberapa ilustrasi bergrafik.
- (25/100)
- (d) Andaikan bahawa anda mencincang kunci  $k$  ke dalam jadual cincangan yang terdiri daripada  $n$  slot (diindekskan dari 0 hingga  $n-1$ ). Bagi setiap fungsi  $h(k)$  berikut, adakah fungsi berkenaan boleh diterima sebagai fungsi cincangan (iaitu adakah fungsi cincangan berkenaan berjalan dengan baik bagi penyisipan dan penggelintaran), dan jika ya, adakah fungsi berkenaan satu fungsi cincangan yang baik? Justifikasikan jawapan anda.
- (i)  $h(k) = k/n$ ,  $k$  dan  $n$  adalah integer.
  - (ii)  $h(k) = 1$ .
  - (iii)  $h(k) = (k + \text{Random}(n)) \bmod n$ , fungsi  $\text{Random}(n)$  mengembalikan satu integer rawak di antara 0 dan  $n-1$ , inklusif.
  - (iv)  $h(k) = k \bmod n$ ,  $n$  adalah nombor perdana.
- (20/100)
- (e) Satu cara untuk mencincang satu rentetan aksara menjadi satu integer ialah dengan mula-mulanya menghasiltambahkan nilai-nilai ASCII setiap aksara dan akhir sekali mengenakan pengendalian modulo ke atas hasil berkenaan.
- (i) Tulis fungsi cincangan berkenaan dalam C++.
  - (ii) Adakah fungsi tersebut suatu fungsi cincangan yang baik? Justifikasikan jawapan anda.
- (20/100)

4. (a) Satu contoh aplikasi graf ialah penjelajahan graf bandar raya. Pengguna aplikasi sedemikian boleh memulakan penjelajahan dari mana-mana bandar raya dan kemudian pergi ke salah satu bandar raya yang boleh dicapai oleh bandar raya semasa. Penjelajahan kemudiannya disambung semula dengan bandar raya yang dipilih itu. Begitulah juga penjelajahan selanjutnya sehingga pengguna keluar dari sistem berkenaan. Bincangkan kesesuaian menggunakan perwakilan matriks kesebelahan, dan perwakilan senarai kesebelahan untuk aplikasi sedemikian.

(20/100)

- (b) (i) Dalam keadaan apakah algoritma penyusuran graf melawat hanya satu subset daripada bucu-bucu graf?  
 (ii) Subset tersebut digelar komponen terkait yang mengandungi  $v$ , dan  $v$  adalah bucu mula penyusuran. Bagaimanakah anda menentukan semua komponen terkait bagi sesuatu graf yang diberikan?  
 (iii) Tentukan semua komponen terkait graf berikut.



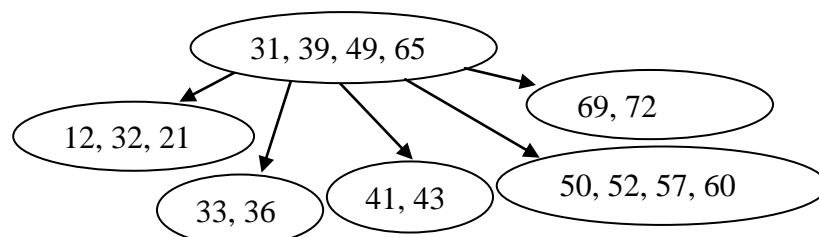
- (iv) Tulis pseudokod C++ untuk menentukan bilangan komponen terkait di dalam sebuah graf yang diberikan. Anda boleh menggunakan fungsi dfs atau bfs (seperti yang diberikan dalam kuliah).

(35/100)

- (c) Sebuah fail luaran mengandungi 12 rekod. Anda ingin mengisih rekod-rekod ini. Ingatan dalaman komputer boleh menampung hanya 3 rekod pada sesuatu masa. Ilustrasikan bagaimana isihan cantum luaran mengisih rekod-rekod berkenaan. Rekod-rekod tersebut adalah: S O A L A N M U D A H A.

(15/100)

- (d) (i) Huraikan algoritma bagi tableRetrieve (capaian kembali jadual – seperti yang diberikan dalam kuliah) bagi pepohon B luaran (Jangan tulis sebarang kod/pseudokod).  
 (ii) Diberikan di bawah pepohon B berdarjah 5. Lukis pepohon B yang terhasil daripada penghapusan 21 daripada pepohon berkenaan.



(30/100)

5. (a) (i) Huraikan kaedah penyuaian pertama dan keadah penyuaian terbaik.  
 (ii) Strategi (kaedah) manakah yang terbaik? (25/100)
- (b) (i) Huraikan format untuk senarai bebas, blok bebas dan apa yang terjadi sebelum dan selepas peruntukan. Masukkan dalam jawapan anda beberapa ilustrasi bergrafik dan bagaimana mendapatkan alamat blok yang baru diperuntukkan.  
 (ii) Mengapakah pelaksanaan senarai berpaut lebih baik daripada pelaksanaan senarai berpaut biasa untuk senarai bebas? (30/100)
- (c) Diberikan algoritma berikut yang mencari satu subrentetan di dalam satu rentetan yang diberikan.

```

int Find (const String& source, const String& target)
// Return the position in source of the first occurrence of
// substring target, and returns -1 if unsuccessful
{ 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
}

```

Di samping analisis secara teori, kita boleh juga melakukan analisis secara empirik. Ubahsuaikan algoritma di atas untuk memasukkan beberapa ukuran yang diperlukan, supaya kita dapat melakukan analisis secara empirik ke atasnya.

(20/100)

- (d) (i) Takrifkan gelintaran menyeluruh.  
 (ii) Secara ringkas, ilustrasikan jawapan anda dalam 5(d)(i) di atas dengan satu masalah yang boleh diselesaikan menggunakan gelintaran menyeluruh.

(25/100)