

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama  
Sidang Akademik 1997/98

September 1997

**CTP201 - Reka Bentuk & Analisis Algoritma**  
**CSC201 - Struktur Data & Algoritma**

Masa : [3 jam]

---

**ARAHAN KEPADA CALON:**

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **LIMA** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
  - Jawab mana-mana **EMPAT** soalan dan semua jawapan hendaklah ditulis dalam Bahasa Malaysia.
- 

...2/-

1. (a) (i) Berikan takrifan **class** dalam C++ bagi *tindanan* yang menggunakan pelaksanaan tatasusunan termasuklah pelaksanaan bagi fungsi-fungsi anggota.
- (ii) Nyatakan manfaat penggunaan **class** jika dibandingkan dengan penggunaan **struct** bagi melaksanakan struktur data tindanan. Huraikan jawapan anda dengan merujuk kepada jawapan yang anda berikan dalam bahagian (i) di atas.
- (iii) Bagaimanakah **template** boleh ditakrif bagi kelas tindanan yang anda berikan dalam bahagian? Nyatakan manfaat penggunaan **template** dalam pelaksanaan struktur data seperti tindanan.

[40/100]

- (b) Berikan takrif secara ber matematik *tatatanda O*

$$g(N) = O(f(n))$$

dan huraikan bagaimana konsep ini berkait dengan idea dalam analisis *kes terburuk*.

[20/100]

- (c) Dua algoritma A dan B menyelesaikan masalah yang sama, A mengambil  $N^2$  saat dan B mengambil N hari.
- (i) Apakah yang dimaksudkan dengan kenyataan di atas dari segi *prestasi relatif* algoritma-algoritma berkenaan?
- (ii) Algoritma manakah lebih baik jika N hanya mengambil nilai *sehingga* 10,000 sahaja?
- (iii) Berapakah sepatutnya nilai N sebelum B mengambil *setengah* daripada masa yang diambil oleh N?

[20/100]

- (d) Berikan analisis *masa larian* dalam *tatatanda O* bagi cebisan atur cara berikut dan tunjukkan bagaimana anda mendapatkan jawapan anda.

```
(i) for (int i = 0, Sum = 0; i < N; i++)
      Sum++;
      for (int j = 0, Sum = 0; j < N; i++)
          Sum++;
```

```
(ii) for (int i = 0, Sum = 0; i < N; i++)
        for (int j = 0, j < N*N; j++)
            Sum++;
```

[20/100]

...3/-

2. (a) Kenalpastikan kaedah(-kaedah) yang *terpantas* antara ketiga-tiga kaedah pengisihan permulaan berikut iaitu isihan sisip, isihan pilih dan isihan gelembung jika fail

- (i) Terisih.
- (ii) Terisih secara terbalik.
- (iii) Terdiri daripada kunci-kunci yang sama nilai.

Justifikasikan jawapan anda.

[20/100]

- (b) (i) Tulis satu fungsi C++, **delete2nd** yang menghapuskan unsur *kedua* terbesar dalam sesebuah timbunan. Fungsi berkenaan boleh menjadi satu *fungsi anggota* tambahan bagi class PQ (baris gilir keutamaan yang diberikan dalam kuliah). Anda boleh menggunakan fungsi-fungsi anggota kelas PQ yang lain untuk melaksanakan fungsi tersebut.

- (ii) Apakah kekompleksan algoritma anda dalam (i) di atas? Jelaskan.

[30/100]

- (b) (i) Dengan menggunakan perkataan anda sendiri, perihalkan kaedah isihan *pertukaran radiks*.

- (ii) Lakukan isihan *pertukaran radiks* bagi fail berikut:

001, 011, 111, 110, 010, 001.

[20/100]

- (d) (i) Mengapakah agaknya algoritma isih cantum bagi isihan dalaman adalah paling sesuai sebagai asas dalam mereka bentuk algoritma bagi kaedah isihan luaran umpamanya cantuman berbilang hala terimbang dan cantuman polifasa?

- (ii) Dalam mereka bentuk algoritma isihan *luaran* kenapakah aspek "sistem" sama pentingnya dengan aspek "algoritma"?

- (iii) Bandingkan prestasi relatif cantuman *berbilang hala terimbang* dan cantuman *polifasa*.

[30/100]

3. (a) Dengan merujuk kepada fungsi-fungsi di bawah, huraikan bagaimana Algoritma Gelintaran Kedalaman Dahulu (perwakilan senarai kesebelahan) boleh *digunakan* atau *diubahsuaikan* untuk menyelesaikan masalah-masalah berikut:
- (i) Menentukan sama ada sesuatu graf itu *terkait*.
  - (ii) Menentukan *komponen-komponen terkait* sesuatu graf.
  - (iii) Mencetak bucu-bucu yang boleh *dicapai* dari sesuatu bucu (graf berarah).
  - (iv) Melakukan isihan topologi *terbalik* (graf berarah).

```

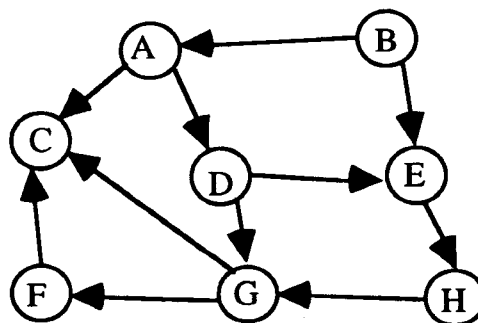
void search()
// Gelintaran
{
    int k;
    for (k = 1; k <= V; k++)
        val[k] = unseen;
    for (k = 1; k <= V; k++)
        if (val[k] == unseen)
            visit(k);
}

void visit(int k)
// Gelintaran Kedalaman Dahulu,
// Senarai kesebelahan
{
    struct node *t;
    val[k] = ++id;
    for (t = adj[k]; t != z; t = t->next)
        if (val[t->v] == unseen)
            visit(t->v);
}

```

[30/100]

- (b) Diberikan graf berarah di bawah.



...5/-

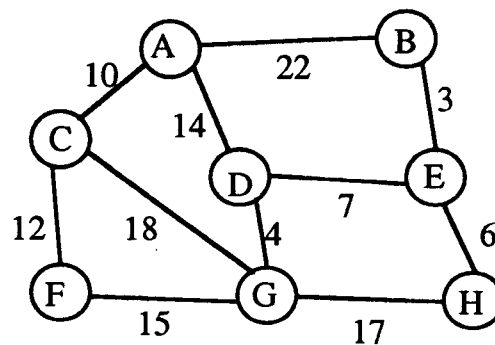
Dapatkan *satu* output bagi setiap isihan berikut untuk graf di atas:

- (i) Isihan topologi.
- (ii) Isihan topologi terbalik.

Tunjukkan langkah demi langkah bagaimana output yang anda berikan diperolehi.

[20/100]

- (c) Pertimbangkan graf berpemberat tak berarah berikut:



- (i) Lakarkan *pepohon rentang minimum* yang berakarakan A, dengan menggunakan *Algoritma Kruskal*. Tunjukkan langkah demi langkah termasuklah kandungan baris gilir keutamaan.
- (ii) Kedua-dua algoritma pepohon rentang minimum yang diberikan dalam kuliah menggunakan konsep membina dari mula sehinggalah sebuah pepohon rentang terhasil. Walau bagaimanapun kita boleh membina pepohon tersebut dari arah yang berlawanan iaitu dengan menghapuskan satu persatu tepi-tepi mengikut nilai pemberat yang maksimum untuk menghapuskan kitar sehinggalah tidak wujud lagi kitar dan dengan itu pepohon rentang yang minimum terhasil. Algoritma ini seolah-olah Algoritma Kruskal yang terbalik. Berikan *rangka* algoritma yang menggunakan pendekatan ini.
- (iii) Tunjukkan langkah demi langkah dan kandungan baris gilir keutamaan dengan menggunakan pendekatan dalam bahagian (ii) di atas bagi graf di atas.

[50/100]

...6/-

4. (a) (i) Dalam *kes terburuk*, gelintaran pada *pepohon gelintaran perduaan* dengan  $N$  kunci mungkin memerlukan  $N$  perbandingan. Huraikan berserta dengan contoh kebenaran kenyataan ini.
- (ii) Kenapakah pentingnya pepohon perduaan *diseimbangkan* untuk memanfaatkan kebaikan yang diberikan oleh struktur ini dalam kebanyakan aplikasi?
- (iii) Berikan huraian ringkas tentang *dua* teknik pengimbangan pepohon yang anda ketahui. Bandingkan secara *relatif* prestasi kedua-dua teknik yang anda berikan.
- (iv) Bermula dengan pepohon AVL yang *kosong*, sisipkan 1, 2, 5, 3, 4, 5, 6, 7 mengikut susunan tersebut. Nyatakan keadaan imbalan pada setiap tahap.

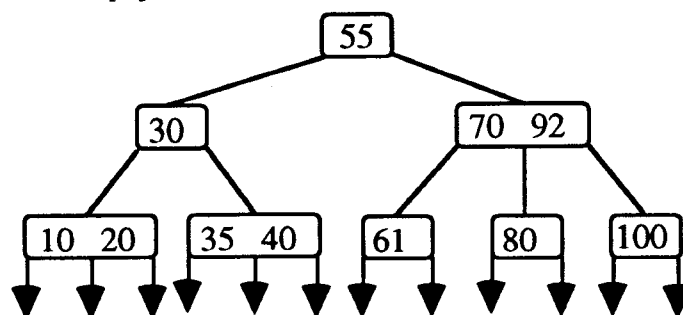
[50/100]

- (b) Enam kunci berikut Al, FL, GA, NC, SC dan VA mestilah dibezakan dengan semua pasangan huruf-huruf yang lain. Untuk menyelesaikan masalah ini, kunci-kunci ini disimpan dalam satu jadual cincang supaya kunci-kunci ini dapat dicari dengan mudah melalui satu fungsi cincangan  $h$ .

- (i) Berikan satu *fungsi cincangan* yang sesuai untuk memetakan set kunci-kunci ini kepada jadual cincang.
- (ii) Jika perwakilan jadual cincang yang digunakan ialah *rantai berasingan*, tulis satu fungsi C++ yang mengembalikan nilai sama ada sesuatu kunci itu *berada* di dalam jadual cincang atau tidak.

[30/100]

- (c) (i) Huraikan istilah *pepohon B*.
- (ii) Diberikan pepohon B *bertertib 3* berikut:



Berikan satu contoh penyisipan (atau satu siri penyisipan) kunci baru yang menyebabkan *pemecahan* nod, dan satu lagi contoh yang menyebabkan *ketinggian* pepohon bertambah.

[20/100]

...7/-

5. (a) Diberikan algoritma berikut yang mencari *kejadian pertama* satu corak p dalam rentetan teks a.

```
int brutesearch(char *p, char *a)
{
    int i, j, M = strlen(p), N = strlen(a);
    for (i = 0, j = 0; j < M && i < N; i++, j++)
        if (a[i] != p[j]) { i -= j-1; j = -1; }
    if (j == M) return i-M; else return i;
}
```

- (i) Huraikan dalam perkataan anda sendiri bagaimana algoritma ini beroperasi. Anda mestilah menyokong jawapan anda dengan menggunakan contoh-contoh yang sesuai.
- (ii) Analisis algoritma di atas dalam situasi *kes terburuk*.
- (iii) Ubahsuaikan supaya algoritma di atas mengimbas corak *dari kanan ke kiri*.
- (vi) Adakah kecekapan algoritma yang anda berikan dalam bahagian (iii) bertambah baik jika dibandingkan dengan algoritma yang asal di atas? Jelaskan.

[35/100]

- (b) Diberikan ungkapan aritmetik berikut dalam bahasa C++.

$$hhh3 = hhh2 * 20.0 + hhh1 * 60;$$

Berikan satu *ringkasan* tentang bagaimana kenyataan ini melalui fasa-fasa pengkompilan/penterjemahan dalam susunan yang diberikan iaitu penganalisis leksikal, penganalisis sintaksis, penganalisis semantik, penjanaan kod pertengahan, dan pengoptimuman kod.

[25/100]

- (c) (i) Terangkan secara ringkas *proses pemadatan storan* dan peranan proses ini dalam teknik *pungutan sampah*.
- (ii) Terangkan pula secara ringkas *proses pemadatan blok storan* dalam konteks pengurusan ingatan dinamik dan peranan proses ini dalam *pengurusan ingatan dinamik*.
- (iii) Berikan contoh satu keadaan yang memerlukan pemadatan blok storan (rujuk bahagian (ii) di atas). Berikan gambar rajah yang menunjukkan keadaan ini dan gambar rajah yang menunjukkan keadaan *selepas* pemadatan dilakukan.

[40/100]