

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 1996/97

Oktober/November 1996

CSC201/CSM301 - Struktur Data & Algoritma

Masa : [3 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **LIMA** soalan di dalam **LAPAN** 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) Huraikan ketiga-tiga pendekatan dalam penganalisan algoritma *secara eksperimen* berikut.
- (i) Menghitung masa CPU.
 - (ii) Menghitung bilangan perbandingan kekunci.
 - (iii) Menghitung bilangan umpukan kekunci.

[25/100]

- (b) Salah satu pengubahsuaian yang boleh dilakukan dalam bahasa C terhadap pelaksanaan sesuatu algoritma isihan ialah dengan menambah satu baris pernyataan untuk menghitung bilangan perbandingan atau umpukan kekunci. Umpamanya, untuk menghitung bilangan perbandingan kekunci :

```

void sort (int list[], int n)
{
    .....;
    count++;
    if (list[i] < list[j])
        .....;
}

```

Terdapat tiga cara untuk berkomunikasi balik kepada atur cara yang memanggil fungsi berkenaan.

- A: Dengan mengubah suai fungsi untuk mengembalikan *nilai secara tersirat*.

```

long sort (int list[], int n)
{
    .....;
    return(count);
}

```

- B: Dengan menghantar nilai berkenaan melalui *senarai parameter*.

```

void sort (int list[], int n, long *count)
{
    .....;
    (*count)++;
}

```

...3/-

C: Dengan menggunakan *pemboleh ubah sejagat*.

```

long count;
main()
{
    .....;
}

void sort (int list[], int n)
{
    .....;
}

```

Cara manakah yang menyebabkan pengubahsuaian yang minimum terhadap fungsi isihan berkenaan dan atur cara yang memanggilnya, dan dengan itu paling mudah dilaksanakan? Justifikasikan jawapan anda dengan mengkritik setiap pendekatan di atas.

[25/100]

(c) Berikut diberikan fungsi bagi kaedah *isihan sisip*:

```

void insertsort(int x[], int n)
{
    int i, k, y;

    for (k = 1; k < n; k++) {
        y = x[k];
        for (i = k - 1; i >= 0 && y < x[i]; i--)
            x[i+1] = x[i];
        x[i+1] = y;
    }
} /* end insertsort */

```

- (i) Ubah suaikan fungsi di atas untuk menghitung bilangan *perbandingan* dan bilangan *umpukan* kekunci dengan menggunakan salah satu pendekatan dalam (b) di atas.
- (ii) Analisis masa komputeran bagi fungsi di atas dengan menggunakan *Hukum Kirchoff* dan memberikan *kes terburuk* dan *kes terbaik*.
- (iii) Jika diberikan senarai data adalah 1, 2, 3, 4, dan 5, dalam turutan berkenaan berapakah bilangan *perbandingan* dan bilangan *umpukan* kekunci yang dilakukan?
- (iv) Jika senarai data adalah dalam *susunan menurun* bagaimanakah anda boleh mengubah suai fungsi di atas supaya algoritma ini lebih cepat? Jelaskan.

[50/100]

...4/-

2. (a) Huraikan *hubungan R* ke atas sesuatu set *A* dan kaitannya dengan graf berarah.

[5/100]

- (b) Jika diberikan hubungan *R* ke atas sesuatu set *A* adalah *jika* $(x > y)$ *dan* $(x - y)$ *adalah ganjil maka* x *berhubungan dengan* y *dan set A adalah* $\{1,2,3,4,5\}$.

- (i) Lakarkan graf berarah bagi hubungan di atas.
- (ii) Tunjukkan perwakilan *senarai kesebelahan* bagi graf di atas.
- (iii) Tuliskan cebisan atur cara dalam bahasa C yang menjanakan graf yang dihasilkan oleh hubungan di atas dalam bentuk *senarai kesebelahan*.
- (iv) Dapatkan *matriks kesebelahan* bagi graf dalam (i) di atas dan tunjukkan langkah demi langkah untuk mendapatkan *matriks tutupan transitif* iaitu untuk mendapatkan $path_k[j][j]$ dengan menggunakan *Algoritma Warshall*.

[50/100]

- (c) Kenapakah perwakilan *senarai multipaut/multisenarai* boleh dianggap sesuai untuk melaksanakan kaedah penjadualan projek *berasaskan penggunaan graf*.

[5/100]

- (d) Bincangkan kecekapan algoritma-algoritma yang terlibat dalam pelaksanaan kaedah penjadualan projek apabila menggunakan *senarai multipaut/multisenarai* dengan beberapa variasi berikut.

- (i) Senarai nod kepala dan senarai lengkok adalah senarai berpaut biasa.
- (ii) Senarai nod kepala dan senarai lengkok adalah senarai berpaut membulat.
- (iii) Senarai nod kepala adalah senarai berpaut ganda dua sementara senarai lengkok adalah senarai berpaut biasa.
- (iv) Senarai nod kepala adalah suatu jadual cincang sementara senarai lengkok adalah senarai berpaut biasa.

[40/100]

...5/-

3. (a) Pertimbangkan algoritma *rekursi* `dftraverse(s)` di bawah bagi *penyusuran kedalamannya dahulu* (Takrifan fungsi-fungsi dan jenis data adalah seperti yang diberikan dalam kuliah).

```

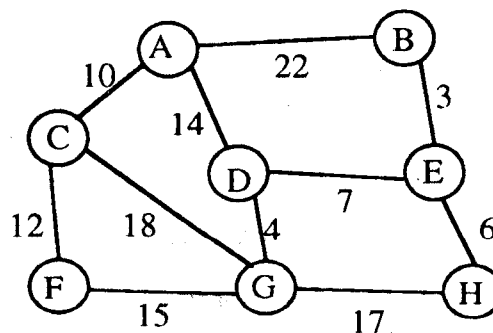
visit(s);
firstsucc(s,yptr,nd);
while (yptr != NULL) {
    if (visited(nd) == FALSE)
        dftraverse(nd);
    nextsucc(s,yptr,nd);
} /* end while */

```

- (i) Terangkan (secara bergambar jika perlu) peranan fungsi-fungsi `visit()`, `visited()`, `firstsucc()` dan `nextsucc()` dalam konteks perwakilan *senarai multipaut/multisenarai*.
- (ii) Huraikan tujuan pernyataan "`if (visited(nd) == FALSE)`" dalam algoritma di atas. Kenapakah pernyataan ini diperlukan sedangkan pernyataan sedemikian tidak diperlukan bagi penyusuran pepohonan perduaan?
- (iii) Kenapakah algoritma di atas perlu diletakkan di dalam sebuah gelung bersama-sama satu fungsi `select()` iaitu fungsi yang mengembalikan nod yang belum dilawati? Kenapakah fungsi ini diperlukan sedangkan fungsi sedemikian tidak diperlukan bagi penyusuran pepohonan perduaan?
- (iv) Ubah suai algoritma di atas supaya satu senarai *isihan bertopologi yang terbalik* dihasilkan. Tunjukkan bagaimana algoritma berkenaan melakukannya.

[60/100]

- (b) Pertimbangkan graf berpemberat tak berarah berikut:



- (i) Gunakan strategi *penyusuran kedalamannya dahulu* untuk menyusur graf di atas, bermula dari bucu A. Berikan satu penyusuran dengan menyenaraikan susunan lawatan dan lakarkan *pepohon rentang kedalamannya dahulu* yang sepadan.

...6/-

- (ii) Lakarkan *pepohon rentang minimum* yang berakarkan A, dengan menggunakan *Algoritma Prim*. Tunjukkan langkah demi langkah.

[30/100]

- (c) Terangkan *perbezaan* antara pepohon rentang kedalaman dahulu dan pepohon rentang minimum.

[10/100]

4. (a) Jika *duplikasi* dibenarkan dalam pepohon gelintaran perduaan (kekunci yang sama nilai dibenarkan) satu konvensi perlu ditetapkan untuk menentukan hubungan antara duplikasi-duplikasi yang ada di dalam struktur data berkenaan. Item yang menduplikasi akar sesuatu pepohon mestilah berada semuanya di subpepohon kiri ataupun semuanya di subpepohon kanan dan ini juga mestilah benar bagi semua subpepohon.

- (i) Kenapakah konvensi ini penting bagi pepohon gelintaran perduaan?
 (ii) Binakan pepohon perduaan yang sedemikian jika data yang diselitkan adalah dalam susunan berikut:

17 8 16 17 12 4 26 4

[20/100]

- (b) Diberikan algoritma pembinaan pepohon gelintaran perduaan yang asal seperti berikut (Takrifan fungsi-fungsi dan jenis data adalah seperti yang diberikan dalam kuliah) :

```

tree = maketree(x[0]);
for (i = 1; i < n; i++) {
  y = x[i];
  q = tree;
  p = q;
  while ( p != NULL ) {
    q = p;
    p = (y < p->k) ? p->left : p->right;
  } /* end while */
  if (y < q->k)
    setleft(q,y);
  else
    setright(q,y);
} /* end for */

```

Ubah suai algoritma ini untuk membenarkan duplikasi data ke dalam pepohon gelintaran perduaan seperti yang disebutkan dalam (a) di atas.

[10/100]

...7/-

(c) Bagi setiap keadaan berikut, tentukan sama ada *teknik pengalamatan terbuka* atau *teknik rantaian* lebih sesuai untuk mengawal limpahan. Justifikasikan jawapan anda

- (i) Bilangan item yang diselitkan mungkin melebihi saiz jadual cincang.
- (ii) Ruang adalah penting, saiz item besar, dan bilangan item sebenar yang dimasukkan adalah kecil daripada saiz jadual cincang.
- (iii) Ruang adalah penting, saiz item kecil, dan bilangan item sebenar yang dimasukkan adalah kecil sedikit daripada saiz jadual cincang.
- (iv) Masa adalah penting, dan bilangan sebenar item yang dimasukkan adalah kecil sedikit daripada saiz jadual cincang.

[30/100]

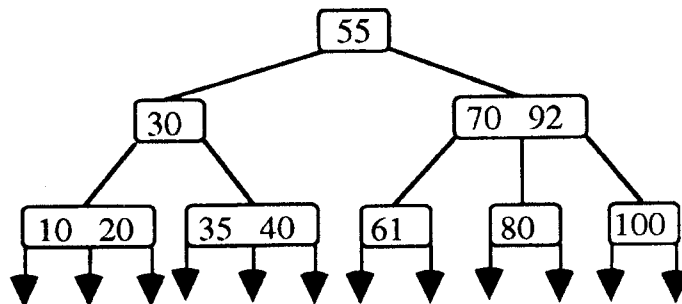
(d) Diberikan fungsi berikut, gunakan cincangan dengan teknik pengawalan limpahan *carian linear* ke dalam jadual cincang berikut:

$h(\text{kekunci}) = \text{kekunci} \% 13$
 (% adalah pengoperasi modulus dalam bahasa C)
 Jujukan input = 10, 12, 16, 29, 21, 25.
 Jadual cincang asal adalah seperti berikut :

0	1	2	3	4	5	6	7	8	9	10	11	12
						19		8		23		

[20/100]

- (e) (i) Terangkan perbezaan antara pepohon B dan pepohon B⁺.
- (ii) Diberikan pepohon B *bertertib 3* berikut:



Bermula dengan pepohon di atas, apakah pepohon B yang terhasil apabila nilai-nilai 15, 37, dan 65 *diselitkan* mengikut turutan tersebut?

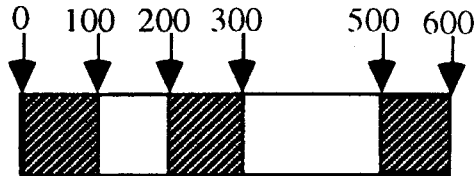
[20/100]

...8/-

5. (a) Huraikan teknik *penyuaian pertama* dan teknik *penyuaian terbaik*.

[15/100]

- (b) Pertimbangkan ingatan di bawah. Ruang kosong menandakan ruang yang bebas dan ruang yang berlorek adalah ruang digunakan.



Berikan satu *jujukan permintaan* yang boleh dipenuhi oleh sistem berkenaan jika

- (i) Teknik *penyuaian pertama* digunakan tetapi permintaan berkenaan tidak dipenuhi jika teknik *penyuaian terbaik* digunakan.
 (ii) Teknik *penyuaian terbaik* digunakan tetapi permintaan berkenaan tidak dipenuhi jika teknik *penyuaian pertama* digunakan

[20/100]

- (c) (i) Terangkan dengan ringkas kenapa algoritma *pungutan sampah* diperlukan.
 (ii) Terangkan dengan ringkas teknik *pungutan sampah*.

[25/100]

- (d) Huraikan peranan *jadual simbol* dalam pengkompil dan *operasi-operasi* yang diperlukan ke atasnya.

[15/100]

- (e) Teknik *cincangan rantaian berasingan* dicadangkan untuk menguruskan jadual simbol dalam sebuah pengkompil. Bincangkan *bentuk* jadual simbol yang sedemikian dan *prestasi masa larian* operasi-operasi yang dilakukan ke atasnya.

[25/100]