

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 1999/2000

Februari 2000

CAP102/CMP102 - Pengaturcaraan Lanjutan & Struktur Data

Masa : [3 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **TIGA** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
 - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. (a) Fungsi Ackermann ialah satu fungsi yang mempunyai dua pemboleh ubah. Diberi definisi formal untuk fungsi tersebut:

$$\begin{aligned} f(0, n) &= n + 1 \\ f(m, 0) &= f(m - 1, 1) \\ f(m, n) &= f(m - 1, f(m, n - 1)) \end{aligned}$$

Tulis satu fungsi rekursi dalam bahasa C untuk melaksanakan fungsi di atas. (5/100)

- (b) Beri dan bincang tentang empat fasa penghalusan data untuk menyelesaikan masalah. (5/100)

- (c) Diberi pengisytiharan struktur data seperti berikut:

```
typedef struct nodgred {
    char nama [25];
    float markah;
    struct nodgred *next;
} NODGRED, *PNODGRED;
```

Tulis pernyataan atau jujukan pernyataan yang akan melaksanakan setiap yang berikut. Anggapkan semua pengolahan berlaku dalam atur cara utama dengan itu tidak perlu alamat pemboleh ubah penuding

- (i) Isytiharkan satu jenis baru bernama **SENARAI** yang mengandungi satu penuding bernama **Kepala** yang menuding kepada **NODGRED**.
- (ii) Menggunakan pengisytiharan dalam (i) di atas, tulis satu fungsi untuk mencipta senarai kosong.
- (ii) Anggapkan senarai yang dituding oleh **Kepala** (diisytiharkan dalam (i) di atas) mempunyai tiga nod. Data yang terkandung dalam nod satu ialah "ANI". Data dalam nod kedua ialah "Salmah" dan data dalam nod ketiga ialah "Zainab". Nod-nod disusun dalam susunan abjad menaik bagi nama. Berikan pernyataan yang akan menyelit nod-nod yang mempunyai data berikut supaya senarai yang terhasil adalah terisih.

<u>Nama</u>	<u>Markah</u>
"Abu"	95.0
"Zarina"	15.0
"Narimah"	50.0

Setiap penyelitan adalah ke dalam senarai asal. Isytiharkan pemboleh-
pemboleh ubah lain jika perlu.

(10/100)

- (d) Diberi prototaip fungsi berikut dalam bahasa C:

```
void StackTop (StackEntry *item, Stack *s)
/* fungsi ini mengembalikan nilai paling atas dalam tindanan tanpa
menggubahkan kandungan asal tindanan */
```

Tulis definisi fungsi StackTop dengan menggunakan pelaksanaan bersebelahan (contiguous). Anda tidak dibenarkan menggunakan fungsi-fungsi asas tindanan.

(5/100)

- (e) Diberi prototaip fungsi untuk operasi-operasi asas ke atas tindanan:

```
void CreateStack (Stack *S);
Boolean StackEmpty (Stack *S);
Boolean StackFull (Stack *S);
void Push (StackEntry item, Stack *S);
void Pop (StackEntry item, Stack *S);
```

Tulis satu algoritma IsihTindanan (Stack *A, Stack *B) yang akan menggabungkan kandungan tindanan A dan tindanan B, menghasilkan satu tindanan baru C yang mempunyai nilai-nilai terisih dengan nilai paling besar berada di atas tindanan.

(10/100)

2. (a) Diberi definisi fungsi Append dan Serve berikut (seperti diberi dalam kuliah):

```
void Append(QueueEntry x, Queue *q)
{
    if (QueueFull(q))
        Error("Cannot append an entry to a full queue.");
    else {
        q -> count ++;
        q -> rear = (q -> rear + 1) % MAXQUEUE;
        q -> entry [q -> rear] = x;
    }
}

void Serve(QueueEntry *x, Queue *q)
{
    if (QueueEmpty(q))
        Error("Cannot serve from an empty queue.");
    else {
        q -> count --;
        *x = q -> entry [q -> front];
        q -> front = (q -> front + 1) * % MAXQUEUE;
    }
}
```

Kita boleh mengimplementasikan dalam bahasa C baris gilir secara membulat yang menyimpan nilai integer dengan menggunakan tatasusunan int Q[SaizQ]. Q[0] mengandungi indeks depan baris gilir dan Q[1] mengandungi indeks belakang senarai Q[2] ke Q[SaizQ-1] mengandungi data.

- (i) Beri pengisytiharan dalam bahasa C untuk implementasi struktur data di atas.
- (ii) Baris gilir kosong apabila (Q[0] == Q[1]), uahsuai fungsi Append dan Serve yang diberi menggunakan struktur data yang anda isytiharkan dalam (i) di atas.

(10/100)

- (b) Diberi algoritma untuk fungsi Gelintaran Kedalaman Dahulu(DFS). Tulis satu fungsi dalam bahasa C yang lengkap menggunakan struktur data abstrak tindanan untuk melaksanakan DFS.

```
void DepthFirst(Graph G, void (*Visit)(Vertex))
{
    Boolean visited [MAXVERTEX];
    Vertex v;
    for (all v in G)
        visited [v] = FALSE;
    for (all v in G)
        if (! visited [v])
            Traverse(v, Visit);
}
```

```
void Traverse(Vertex v, void (*Visit)(Vertex))
{
    Vertex w;
    visited[v] = TRUE;
    Visit(v);
    for (all w adjacent to v)
        if (! visited [w])
            Traverse(w, Visit);
}
```

(5/100)

- (c) Diberi fungsi gelintaran berjujukan berikut:

```
int SequentialSearch (List L,Keytype target)
{
    int location = 0;
    L.entry [L.count].key = target;
    while (L.entry[location].key != target)
        location ++;
    if (location < L.count)
        return location;
    else
        return -1;
}
```

- (i) Bandingkan fungsi di atas dengan fungsi gelintaran berjjukan yang diberi dalam kuliah. Apakah kelebihanannya?
- (ii) Apakah perbezaan antara kaedah gelintaran berjjukan dan gelintaran perduaan?

(5/100)

- (d) Diberi jujukan nombor-nombor berikut:

27 34 36 20 28 11 21

Isih nombor-nombor di atas menggunakan algoritma-algoritma berikut. Surih langkah-demi langkah.

- (i) Isihan Shell
- (ii) Isihan Cepat
- (iii) Isihan Pilih.

(6/100)

- (e) Berapakah perbandingan yang diperlukan bagi setiap algoritma di 2(d).

(4/100)

3. (a) (i) Berikan pengisytiharan dalam bahasa C bagi struktur data nod pepohon gelintaran perduaan yang mengandungi maklumat seperti berikut:

nama, jantina, umur dan kad pengenalan.

- (ii) Tulis satu fungsi **Kira** () dalam bahasa C untuk mengira bilangan nod yang lebih besar daripada kunci **X** yang menyimpan nilai umur di dalam pepohon gelintaran perduaan. Guna struktur data yang diberikan dalam (i) di atas.
- (ii) Tulis suatu fungsi **HapusDaun** () dalam bahasa C yang akan menghapuskan kesemua nod daun. Guna struktur data yang diberikan dalam (i) di atas.

(13/100)

- (b) (i) Bina pepohon perduaan bagi senarai nombor berikut:

100 88 76 52 12 93 18 1 22 7 34 46

- (ii) Tunjukkan hasil penyusunan mengikut kaedah berikut terhadap pepohon dalam (i) di atas:

- Tertib awalan
- Tertib akhiran

(iii) Andaikan jujukan operasi berikut dilaksanakan. Lukis hasil penghapusan bagi setiap operasi terhadap pepohon dalam (i) di atas.

- nod 100 dihapuskan daripada pepohon
- nod 32 diselitkan kepada pepohon

(12/100)

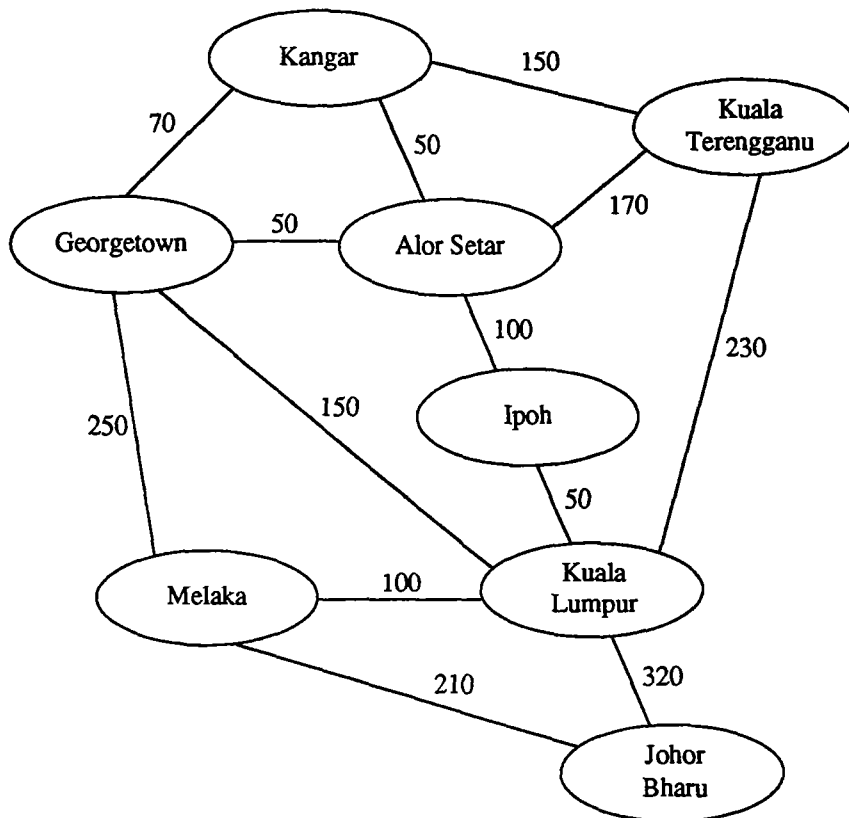
(c) Diberi fungsi Gelintaran Pepohon berikut:

```
TreeNode *TreeSearch (TreeNode *root, KeyType Target)
{
    if (root)
        if (LT (target, root -> entry.key))
            root = TreeSearch (root -> left, target)
        else if (GT (target, root -> entry.key))
            root = TreeSearch (root -> right, target);
        return root;
}
```

Tulis semula fungsi di atas (dalam bahasa C) menggunakan kaedah lelaran (iterative).

(5/100)

(d) Diberikan graf di bawah:



Tunjukkan langkah demi langkah bagaimana untuk mendapatkan laluan terpendek bermula dari nod Kanger ke nod-nod lain menggunakan algoritma TAMAK. Tunjukkan dengan jelas kandungan tatasusunan COST dan DISTANCE.

(5/100)

- oooOooo -