

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 1998/99

Februari 1999

CAP102/CMP102 - Pengaturcaraan Lanjutan Dan Struktur Data
CSC122 - Penyelesaian Masalah Dan Pengaturcaraan

Masa : [3 jam]

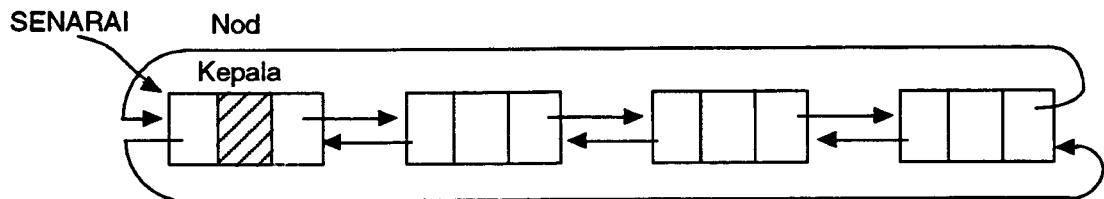
ARAHAN KEPADA CALON:

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

1. (a) (i) Berikan definisi dan contoh struktur data abstrak (Abstract Data Type).
 (ii) Bandingkan implementasi berpaut (linked) dan berdampingan (contiguous).
 (5/25)
- (b) (i) Surih pelaksanaan fungsi di bawah dengan menunjukkan kandungan tindanan pada setiap langkah surihan apabila fungsi tersebut dipanggil dengan Teka (2, 5).

```
int Teka (int m, int n)
{
  if (m < n) {
    return m*m + Teka (m + 1, n);
  } else {
    return m*m;
  }
}
```

- (ii) Tulis semula fungsi rekursi di atas dengan menggunakan kaedah lelaran (iterative).
 (iii) Kaedah manakah yang lebih sesuai digunakan untuk fungsi dalam (i) di atas - rekursi atau lelaran? Jelaskan jawapan anda.
 (8/25)
- (c) Pertimbangkan senarai ganda dua membulat di bawah.



- (i) Berikan pengisytiharan bahasa C untuk senarai di atas untuk:
- implementasi berdampingan
 - implementasi berpaut
- (ii) Tulis pernyataan bahasa C yang akan melaksanakan operasi berikut dengan menggunakan implementasi berpaut.
- Menyisip nod baru ke senarai ganda dua membulat yang kosong.
 - Mengira jumlah nod yang terdapat dalam senarai ganda dua membulat.
 - Menggantikan nilai kesemua nod yang mempunyai nilai X dengan nilai Y.

(12/25)

2. (a) Suatu tindakan berjenis integer boleh dibina dengan menggunakan tatasusunan `int S[MAX_ELEMENT]`. `S[MAX_ELEMENT - 1]` digunakan untuk menyimpan indeks elemen teratas tindakan berkenaan dan `S[0]` hingga `S[MAX_ELEMENT - 2]` mengandungi elemen tindakan sebenar .

- (i) Berikan pengisytiharan yang sesuai untuk mengimplementasi tindakan di atas.
(ii) Tulis fungsi `Push ()` dan `Pop ()` bagi tindakan tersebut.

(8/25)

- (b) Suatu cara lain membina baris gilir secara berpaut ialah dengan menggunakan senarai berpaut membulat yang mempunyai penuding yang menuding kepada nod terakhir baris gilir.

- (i) Adakah cara ini merupakan cara yang cekap. Jelaskan jawapan anda.
(ii) Tulis fungsi bagi setiap operasi baris gilir di bawah:
- Menyemak sama ada baris gilir kosong atau penuh.
 - Operasi **Append** (menyisipkan nod ke belakang garis gilir).

(8/25)

- (c) Contoh aplikasi tindakan yang telah anda pelajari ialah menukar ungkapan tertib sisipan kepada ungkapan tertib akhiran. Tulis fungsi **TukarKeAkhiran** (`char sisipan []`, `char akhiran []`) yang menggunakan prototaip fungsi di bawah dan operasi asas tindakan yang telah anda pelajari.

```
/* Fungsi yang menentukan sama ada c adalah operator */
int AdakahOperator(char c);
```

```
/* keutamaan operator mana yang lebih tinggi, operator1 atau operator 2 */
int Keutamaan (char operator1, char operator2);
```

(5/25)

- (d) Satu baris gilir boleh diimplementasikan menggunakan implementasikan berdampingan. Tulis fungsi di bawah untuk implementasi yang dinyatakan.

- (i) `Boolean QueueEmpty(Stack *S)`
(ii) `Boolean QueueFull(Stack *S)`

(4/25)

3. (a) (i) Beri perbezaan antara pepohon perduaan (binary tree) dan pepohon gelintaran perduaan (binary search tree).

(ii) Lukis pepohon gelintaran perduaan untuk jujukan nombor-nombor berikut:

41 51 6 7 9 18 3 16 9 5

(5/25)

(b) Kita boleh mengisytiharkan satu nod di dalam pepohon dengan menggunakan penuding untuk menuding anak paling kiri dan satu lagi penuding ke adik-beradik.

(i) Berikan pengisytiharan dalam bahasa C untuk implementasi di atas.

(ii) Tulis satu algoritma untuk mencari nod yang mempunyai nilai x bagi implementasi dalam (i).

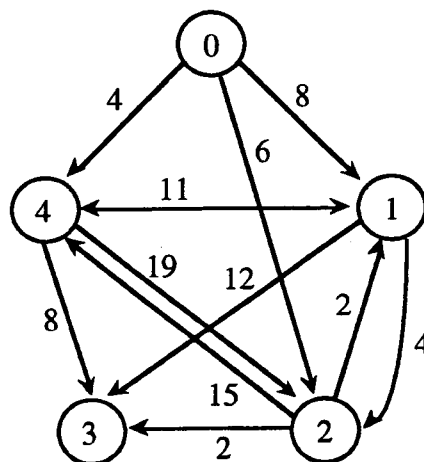
(6/25)

(c) Satu kaedah gelintaran berjujukan yang efisien melibatkan penyisipan kunci yang dicari di hujung tatasusunan sebelum gelintaran, ini akan menyebabkan kunci akan sentiasa dijumpai.

Tulis satu fungsi C untuk mengimplementasikan gelintaran berjujukan di atas.

(5/25)

(d) Diberikan graf di bawah:



- (i) Beri implementasi graf sebagai
- Jadual kesebelahan.
 - Implementasi hibrid antara berdampingan dan berpaut.
- (ii) Tunjukkan langkah demi langkah untuk mendapatkan laluan terpendek menggunakan algoritma Tamak.
- (iii) Tunjukkan langkah demi langkah untuk menjejak graf menggunakan kaedah Gelintaran Kedalaman Dahulu (DFS) bermula dari nod 0. Tunjukkan kandungan tindakan pada setiap langkah.

(9/25)

4. (a) Diberikan:

/ InsertionSort: isih senarai berdampingan melalui kaedah isihan Sisip
 Pre: Senarai kesebelahan telah dicipta. Setiap kemasukan mempunyai kunci.
 Post: Senarai telah diisih menaik */*

```
void InsertionSort(List *list)
{
    Position fu;           /* posisi unsur pertama yang tak diisih */
    Position place;       /* cari bahagian yang terisih yang dikeluarkan dari
                          senarai */
    ListEntry current;    /* menyimpan data yang dikeluarkan dari senarai
                          sementara */

    for (fu = 1; fu < list ->count; fu++)
        if (LT(list->entry[fu].key, list->entry [fu - 1].key)) {
            current = list->entry [fu];
            for (place = fu - 1; place >= 0; place --) {
                list->entry [place + 1] = list->entry [place];
                if (place == 0 || LE (list->entry [place - 1].key,current.key))
                    break;
            }
            list->entry [place] = current;
        }
}
```

Apakah ubahsuaian yang perlu dibuat supaya fungsi di atas menyusun item secara menurun.

(4/25)

- (b) (i) Diberi senarai yang mempunyai 12 unsur.

30	62	53	42	17	97	91	38
60	51	18	96				

Isih nombor-nombor di atas dengan menggunakan mana-mana **tiga** algoritma di bawah:

- Isihan Shell
- Isihan Cepat
- Isihan Pilih
- Isihan Sisip

- (ii) Jelas kenapa memilih nombor perdana untuk nilai tambah dalam Isihan Shell lebih elok.

(8/25)

- (c) Isihan Cepat dan Isihan Cantum adalah algoritma bahagi dan tawan. Beri perbezaan antara kedua-dua kaedah tersebut.

(2/25)

- (d) Pertimbangkan fungsi di bawah:

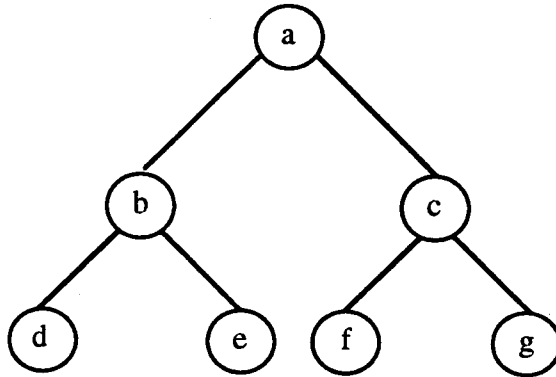
```
void Preorder (Tree Node *root, void (*Visit) (TreeEntry x))
{
    if (root) {
        Visit (root → entry)
        Preorder (root → left, Visit);
        Preorder (root → right, Visit);
    }
}
```

Tulis semula fungsi di atas tanpa menggunakan rekursi. Gunakan struktur data abstrak tindanan.

(7/25)

(e) Tulis satu fungsi dalam Bahasa C yang akan menjejak pepohon paras demi paras.

Contoh:



Output: a b c d e f g

Anda dibenar menggunakan mana-mana struktur data abstrak yang anda telah pelajari.

(4/25)

- oooOOooo -