

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua

Sidang 1986/87

CSP102 - Algoritma & Struktur Data I

Tarikh: 5 April 1987

Masa: 9.00 pagi - 12.00 t/hari
(3 jam)

Sila pastikan bahawa kertas peperiksaan ini mengandungi 9 mukasurat yang tercetak sebelum anda memulakan peperiksaan ini.

Kertas ini mengandungi LIMA soalan. Jawab SOALAN 1 dan mana-mana TIGA soalan lain.

Semua soalan MESTI dijawab dalam Bahasa Malaysia.

Semua algoritma, fungsi dan tatacara hendaklah ditulis dalam Pascal.

1. (a) (i) Senaraikan ciri-ciri penting yang terdapat pada sesuatu algoritma.
- (ii) Berikan satu contoh algoritma yang mempunyai semua ciri-ciri yang anda senaraikan dalam (i) dan tunjukkan bagaimana ciri-ciri tersebut dipenuhi.

(15/100)

- (b) Setengah operasi yang dibenarkan ke atas senarai linear lebih baik dijalankan dengan perwakilan senarai berpaut dan setengah-setengahnya pula lebih baik dengan perwakilan susuntertib. Jelaskan perwakilan yang manakah lebih baik untuk operasi-operasi berikut:

- (i) Penghapusan/penyelitan sesuatu nod.
- (ii) Capaian nod.
- (iii) Carian untuk sesuatu nod.
- (iv) Penggabungan dua senarai/penceraiian kepada dua senarai.

(15/100)

- (c) Diberi a dan b adalah integer positif. Sekiranya suatu fungsi Q ditakrif secara rekursi seperti berikut:

$$Q(a,b) = \begin{cases} 0 & \text{jika } a < b \\ Q(a-b,b)+1 & \text{jika } a \geq b \end{cases}$$

- (i) Cari Q(2,3) dan Q(11,3).
- (ii) Tulis suatu fungsi rekursi untuk fungsi Q sebagaimana yang ditakrifkan di atas.

(15/100)

- (d) Tulis suatu fungsi untuk menghitung bilangan unsur-unsur yang terdapat dalam setiap struktur berikut:

- (i) SetN, jika diberi
TYPE
SetNombor = SET OF 10..25;
VAR
SetN:SetNombor;

- (ii) Senarai berpaut membulat jika penunjuk PJK menunjuk kepada nod semasa senarai berkenaan.

(20/100)

(e) Tulis satu catatan ringkas mengenai setiap perkara berikut:

- (i) Takrif rekursi untuk pohon.
- (ii) Pohon dedua.
- (iii) Pohon dedua lengkap dan pohon dedua wajar lengkap.

(15/100)

(f) (i) Kenapakah sesuatu senarai itu mestilah sedia terisih jika kaedah Carian Dedua digunakan untuk mencari sesuatu nod?

(ii) Diberikan suatu senarai yang terdiri dari integer-integer positif dalam susunan berikut:

2 3 9 18 20 21

Illustrasikan, langkah demi langkah, bagaimana algoritma Carian Dedua boleh digunakan untuk mencari nombor 18 dalam senarai ini.

(20/100)

2. (a) Tulis satu catatan ringkas mengenai perbezaan antara struktur-struktur berikut:

- (i) susunertib dan fail (berjujukan).
- (ii) set dan rekod.

(15/100)

(b) Tulis suatu tatacara untuk setiap operasi berikut:

- (i) Menghapus unsur yang terakhir dalam senarai berpaut.
- (ii) Menghapus unsur yang pertama dalam senarai berpaut.

(Anggapkan penunjuk DEPAN menunjuk kepada unsur yang pertama dalam senarai berkenaan).

(25/100)

- (c) (i) Bagaimanakah senarai berpaut ganda dua boleh dilaksanakan dalam Pascal?
- (ii) Tuliskan satu algoritma untuk menyelitkan suatu nod baru di sebelah kiri nod yang ditunjuk oleh nod X dalam senarai berpaut ganda dua. (Anggapkan nod baru ini ditunjuk oleh penunjuk T).

(25/100)

- (d) Bincangkan kebaikan dan keburukan, jika ada, pada senarai berpaut ganda dua jika dibandingkan dengan senarai berpaut biasa (satu paut) untuk operasi-operasi berikut:

- (i) Merentas senarai untuk memproses setiap unsur.
- (ii) Menghapuskan sesuatu nod yang diberi nilai kunci.
- (iii) Mencari sesuatu nod dalam senarai yang sedia terisih.
- (iv) Mencari sesuatu nod dalam senarai yang tidak terisih.
- (v) Menyelit suatu nod baru sebelum suatu nod yang diberi nilai kunci.
- (vi) Menyelit suatu nod baru selepas suatu nod yang diberi nilai kunci.

(35/100)

3. (a) Apakah ciri-ciri stek, giliran dan giliran dua hujung? Berikan contoh-contoh yang sesuai untuk mengillustrasikan jawapan anda.

(25/100)

- (b) Bagaimanakah anda melaksanakan giliran dalam Pascal dengan menggunakan

- (i) rekod dan susuntertib.
- (ii) rekod dan penunjuk (senarai berpaut).

(20/100)

(c) Pilih dan jelaskan pelaksanaan yang paling sesuai dari (b) untuk menjalankan setiap operasi berikut. Tuliskan suatu algoritma dengan menggunakan pelaksanaan yang anda pilih itu untuk setiap operasi berkenaan.

- (i) Menyelitkan unsur baru di belakang giliran.
- (ii) Menggabungkan dua giliran supaya depan giliran kedua disambungkan ke belakang giliran pertama.

(25/100)

(d) (i) Kenapakah stek digunakan dalam algoritma penilaian ungkapan postfix?

(ii) Tunjukkan, langkah demi langkah, bagaimana penilaian ungkapan postfix berikut dilakukan:

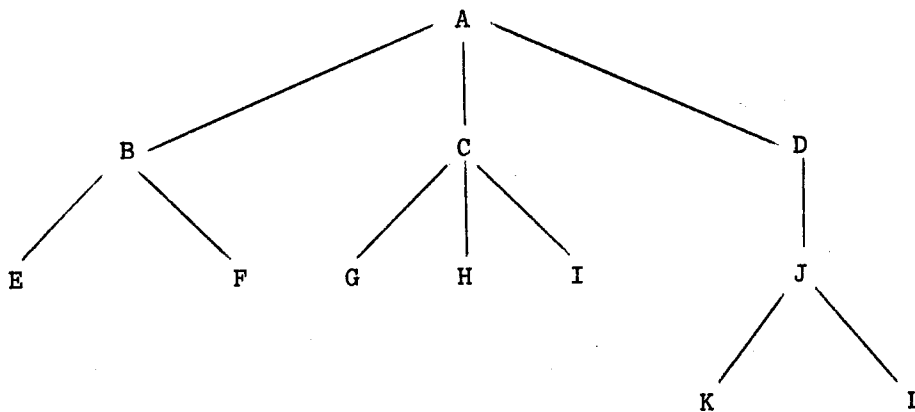
A B C D ↑↑ + E F G / + *

Format jawapan anda mestilah dalam bentuk berikut:

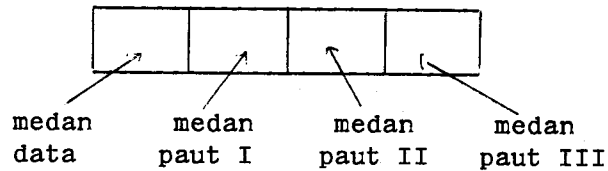
<u>Input</u>	<u>Kandungan Stek</u>	<u>Penilaian</u>	<u>Pertengahan</u>
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:

(30/100)

4. (a) Diberikan suatu pohon am seperti di bawah:



- (i) Tuliskan lintasan antara akar pohon ini dan nod K.
- (ii) Suatu pohon am boleh diwakilkan dengan menggunakan perwakilan berpaut iaitu nod-nod dalam perwakilan ini mempunyai satu medan data dan n medan paut. Nod untuk pohon di atas boleh digambarkan seperti berikut:



Setiap medan paut menunjuk kepada anak-anaknya (satu paut untuk setiap anak) atau NIL.

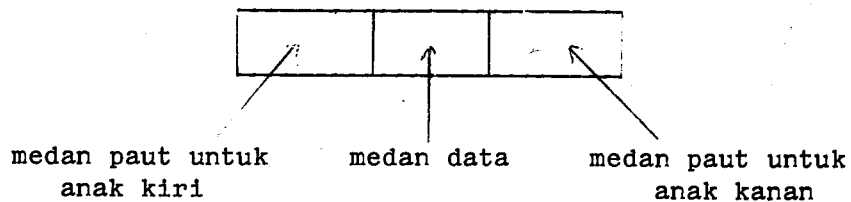
Soalan:

Lakarkan pohon am di atas untuk menunjukkan pautan-pautan yang wujud jika perwakilan ini digunakan.

- (iii) Apakah kebaikan dan keburukan perwakilan dalam (ii), jika ada?

(30/100)

- (b) (i) Transformasikan pohon am dalam (a) ke pohon dedua. (Perhatian: Tunjukkan langkah demi langkah proses ini).
- (ii) Pohon dedua boleh diwakilkan dengan menggunakan perwakilan berpaut iaitu nod-nod dalam perwakilan ini mempunyai satu medan data dan dua medan paut. Satu medan paut menunjuk kepada anak kiri atau NIL dan satu medan lagi menunjuk kepada anak kanan atau NIL. Nod untuk pohon yang menggunakan perwakilan ini boleh digambarkan seperti berikut:



Soalan:

Jika pohon dedua yang dihasilkan dalam (i) diwakilkan dengan perwakilan ini, apakah output yang dihasilkan jika tatacara di bawah digunakan untuk merentas pohon tersebut?

```
PROCEDURE Rentas (P:PJK);
BEGIN
  IF P < >NIL THEN
    BEGIN
      Rentas (P@.Anaknanan);
      Rentas (P@.Anakkiri);
      Writeln (P@.Data)
    END
  END;
END;
```

(30/100)

- (c) Rentasan preorder pohon dedua T menghasilkan nod-nod dalam susunan berikut:

$* + a - b c / - d e - + f g h$

Rentasan inorder pohon dedua T pula menghasilkan nod-nod dalam susunan berikut:

$a + b - c * d - e / f + g - h$

- (i) Dengan menggunakan maklumat di atas, lakarkan pohon dedua T ini.
- (ii) Senaraikan ungkapan infix berserta dengan kurungan (parentesis), jika ada, dengan berdasarkan kepada pohon dedua T ini.
- (iii) Senaraikan juga ungkapan postfix dan ungkapan prefix dengan berdasarkan kepada pohon dedua T ini.

(40/100)

5. (a) Tulis satu catatan ringkas mengenai setiap perkara berikut:

- (i) Kaedah isihan yang mantap (stabil).
- (ii) Timbunan.
- (iii) Analisis kaedah isihan.
- (iv) Operasi-operasi asas untuk rentetan.

(20/100)

(b) Berikut ialah suatu tatacara untuk pengisihan unsur-unsur dalam suatu senarai.

(Anggapkan senarai = ARRAY [0.. MakSen] OF integer;)

```
PROCEDURE Isihxxx(VAR S:senarai;n:integer);
VAR
  I, J, Z : integer;
BEGIN
  FOR I:= 2 to n DO
    BEGIN
      Z := S[I];
      J := I-1 ;
      S [0] := Z :
      WHILE Z < S[J] DO
        BEGIN
          S[J+1]:= S[J];
          J := J-1
        END;
      S [J+1]:= Z;
    END;
  END;
```

- (i) Adakah teknik pengisihan ini mantap? Jelaskan berserta dengan contoh.
- (ii) Anggarkan purata bilangan perbandingan/bilangan anjakan rekod-rekod untuk kaedah ini.

(30/100)

(c) (i) Dirikan suatu timbunan untuk nilai-nilai berikut:

50 20 60 30 80 70

(ii) Dengan menggunakan timbunan yang dihasilkan dalam (i), tunjukkan langkah demi langkah bagaimana algoritma IsihTimbun mengisih nilai-nilai tersebut.

(30/100)

(d) Diberikan rentetan R1 dan R2 seperti berikut:

R1 = 'INI□SUATU□RENTETAN'
R2 = 'INI□SATU□LAGI□RENTETAN'

(Perhatian: □ melambangkan aksara ruang)

Nyatakan hasil bagi setiap operasi berikut, jika operasi-operasi dan fungsi-fungsi primitif rentetan SUBSTR, LENGTH, INDEX dan || adalah seperti yang ditakrifkan dalam kuliah.

- (i) SUBSTR(R1,10,5);
- (ii) LENGTH(R1||SUBSTR(R2,INDEX(R1,'□R')));
- (iii) INDEX(R2,SUBSTR(R1,11));
- (vi) SUBSTR(R2,LENGTH(SUBSTR(R2,15)),3);

(20/100)

...ooOoo...

