UNIVERSITI SAINS MALAYSIA

Second Semester Examination
2009/2010 Academic Session

April/May 2010

MSG 383 – Data Structures for Computer Graphics
[Struktur Data untuk Grafik Komputer]

Duration : 3 hours
[Masa : 3 jam]

Please check that this examination paper consists of ELEVEN pages of printed material before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi SEBELAS muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]

Instructions : Answer **all nine** [9] questions.

*[**Arahan** : Jawab **semua sembilan** [9] soalan.]*

The question papers shall not be taken out from the examination hall and will be collected by invigilators.

Kertas soalan ini tidak boleh dibawa keluar daripada dewan peperiksaan dan akan dikutip oleh pengawas peperiksaan.

In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi Bahasa Inggeris hendaklah diguna pakai].

1. Mark the following statements as "True" or "False".
   (a) In a linked list, the positions of the elements are determined by the order in which the nodes were created.
   (b) In a linked list, memory allocated for the nodes is sequential.
   (c) A singly-linked-list can be traversed in either direction.
   (d) In a linked list, the nodes are always inserted either in the beginning or at the end because a linked list is not a random access data structure.
   (e) The head pointer of a linked list should not be used to travers the list.

   [10 marks]

2. Consider the following recursive function:

```
int mystery (int number)                          // Line 1
{
  if (number == 0)                                // Line 2
    return number;                                // Line 3
  else                                            // Line 4
    return(number + mystery(number-1));           // Line 5
}
```

   (a) Identify the base case(s).
   (b) Identify the induction step(s).
   (c) What valid values can be passed as parameters to the function `mystery`?
   (d) If `mystery(0)` is a valid call, what is its value? If not, explain why.
   (e) If `mystery(5)` is a valid call, what is its value? If not, explain why.
   (f) If `mystery(-3)` is a valid call, what is its value? If not, explain why.

   [10 marks]

3. Mark the following statements as "True" or "False".
   (a) A linear search of a list assumes that the list is in ascending order.
   (b) A binary search of a list assumes that the list is sorted.
   (c) A binary search is faster on sorted lists and slower on unsorted lists.
   (d) A binary search is faster on large lists, but a linear search is faster on small lists.

   [10 marks]

1. *Tandakan pernyataan-pernyataan di bawah sebagai "True" atau "False".*
   (a) *Bagi sesuatu senarai berpaut, kedudukan unsur-unsur ditentukan mengikut susunan nod-nod semasa diciptakan.*
   (b) *Bagi sesuatu senarai berpaut, peruntukan ingatan untuk nod-nodnya dilakukan secara berjujukan.*
   (c) *Senarai berpaut satu-per-satu boleh disusuri dari kedua-dua arah.*
   (d) *Bagi sesuatu senarai berpaut, selitan nod-nod selalunya berlaku di bahagian hadapan ataupun belakang kerana senarai berpaut bukan struktur data yang bersifat capaian rawak.*
   (e) *Penuding kepala untuk sesuatu senarai berpaut tidak harusnya digunakan untuk menyusuri senarai tersebut.*

   [10 markah]

2. *Pertimbangkan fungsi rekursi di bawah*

```
int mystery (int number)                          // Line 1
{
  if (number == 0)                                // Line 2
    return number;                                // Line 3
  else                                            // Line 4
    return(number + mystery(number-1));           // Line 5
}
```

   (a) *Nyatakan unsur(-unsur) asas.*
   (b) *Nyatakan unsur(-unsur) langkah induktif.*
   (c) *Apakah nilai-nilai sah yang boleh digunakan sebagai parameter untuk panggil fungsi* `mystery`*?*
   (d) *Jika* `mystery(0)` *adalah panggilan yang sah, apakah nilainya? Jika tidak sah, jelaskan.*
   (e) *Jika* `mystery(5)` *adalah panggilan yang sah, apakah nilainya? Jika tidak sah, jelaskan.*
   (f) *Jika* `mystery(-3)` *adalah panggilan yang sah, apakah nilainya? Jika tidak sah, jelaskan.*

   [10 markah]

3. *Tandakan pernyataan-pernyataan di bawah sebagai "True" atau "False".*
   (a) *Gelintaran berjujukan bagi sesuatu senarai menganggapkan senarai tersebut berada dalam susunan menokok.*
   (b) *Gelintaran perduaan bagi sesuatu senarai menganggapkan senarai tersebut adalah terisih.*
   (c) *Gelintaran perduaan adalah lebih cepat apabila dilaksanakan ke atas senarai-senarai yang terisih dan lebih lambat ke atas senarai-senarai yang tidak terisih.*
   (d) *Gelintaran perduaan adalah lebih cepat apabila dilaksanakan ke atas senarai-senarai yang besar, tetapi gelintaran berjujukan adalah lebih cepat apabila dilaksanakan ke atas senarai-senarai yang kecil.*

   [10 markah]

…4/-

4. Consider the following program segment,

```
:
int list[] = {2, 10, 17, 45, 49, 55, 68, 85, 92, 98, 110};
int length=11;

int binarySearch (int item)
{
  int first = 0;
  int last = length-1;
  int mid;

  bool found=false;

  while (first<=last && !found) {
    mid = (first+last)/2;
    if (list[mid]==item) found = true;
    else if (list[mid]>item) last = mid-1;
    else first = mid+1;
  }
  if (found) return mid;
  else return -1;
}
:
```

Assume that all variables are declared correctly and there is no syntax error in the program. By using the binary search algorithm above, how many comparisons are required to find whether an item is in `list`? Show the index values of `first`, `last`, `middle` and the number of comparisons after each iteration of the loop. Find the answers for items (a) to (d) by using the table below,

Iteration	`first`	`last`	`mid`	`list[mid]`	Number of comparisons
:	:	:	:	:	:

(a) 15
(b) 49
(c) 98
(d) 99

[10 marks]

4. *Pertimbangkan segmen program berikut,*

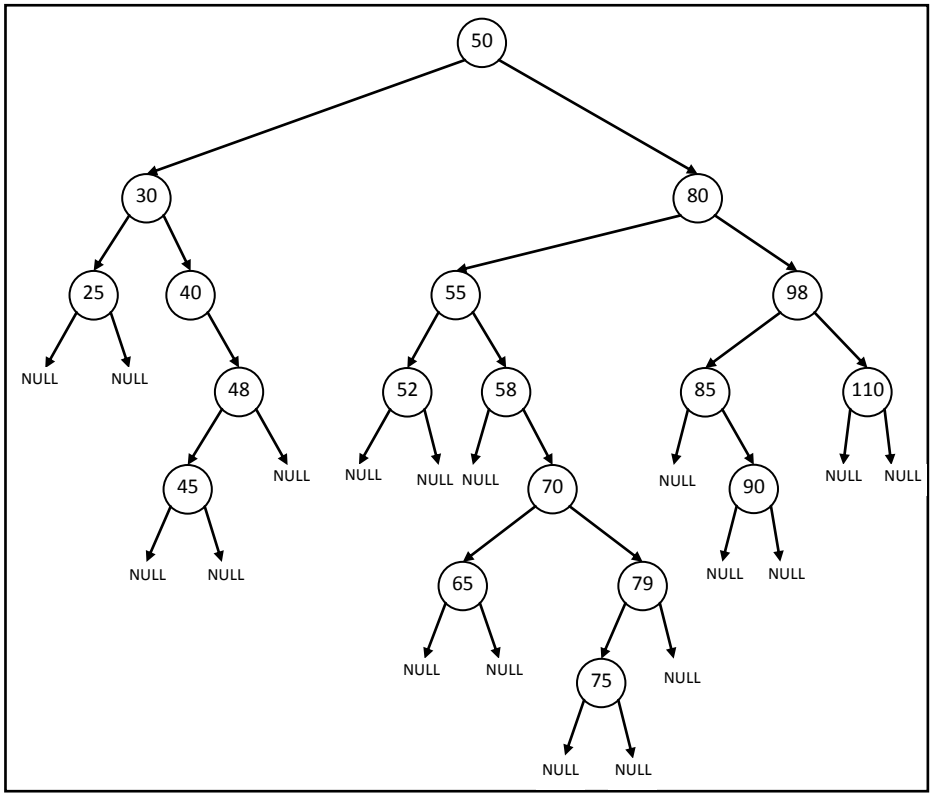
```
:
int list[] = {2, 10, 17, 45, 49, 55, 68, 85, 92, 98, 110};
int length=11;

int binarySearch (int item)
{
  int first = 0;
  int last = length-1;
  int mid;

  bool found=false;

  while (first<=last && !found) {
    mid = (first+last)/2;
    if (list[mid]==item) found = true;
    else if (list[mid]>item) last = mid-1;
    else first = mid+1;
  }
  if (found) return mid;
  else return -1;
}
:
```

Anggapkan semua pembolehubah telah diisytiharkan dengan betulnya dan tiada ralat sintaks dalam program. Dengan menggunakan algoritma gelintaran perduaan di atas, berapakah perbandingan yang dikehendaki untuk mencari samada sesuatu perkara wujud dalam senarai `list`*? Tunjukkan nilai-nilai indeks bagi* `first`*,* `last`*,* `middle` *dan bilangan perbandingan selepas setiap lelaran. Cari jawapan anda untuk item-item dari (a) ke (d) dengan menggunakan jadual di bawah,*

Iteration	`first`	`last`	`mid`	`list[mid]`	Number of comparisons
:	:	:	:	:	:

(a) 15
(b) 49
(c) 98
(d) 99

[10 markah]



5. Refer to the binary search tree above,
   (a) List the path from node with label 80 to the node with label 79.
   (b) A node with label 35 is to be inserted in the tree. Redraw the binary search tree after the insertion.
   (c) Delete node with label 52 and redraw the binary search tree.
   (d) Delete node with label 40 and redraw the binary search tree.
   (e) Delete node with label 80 and redraw the binary search tree.

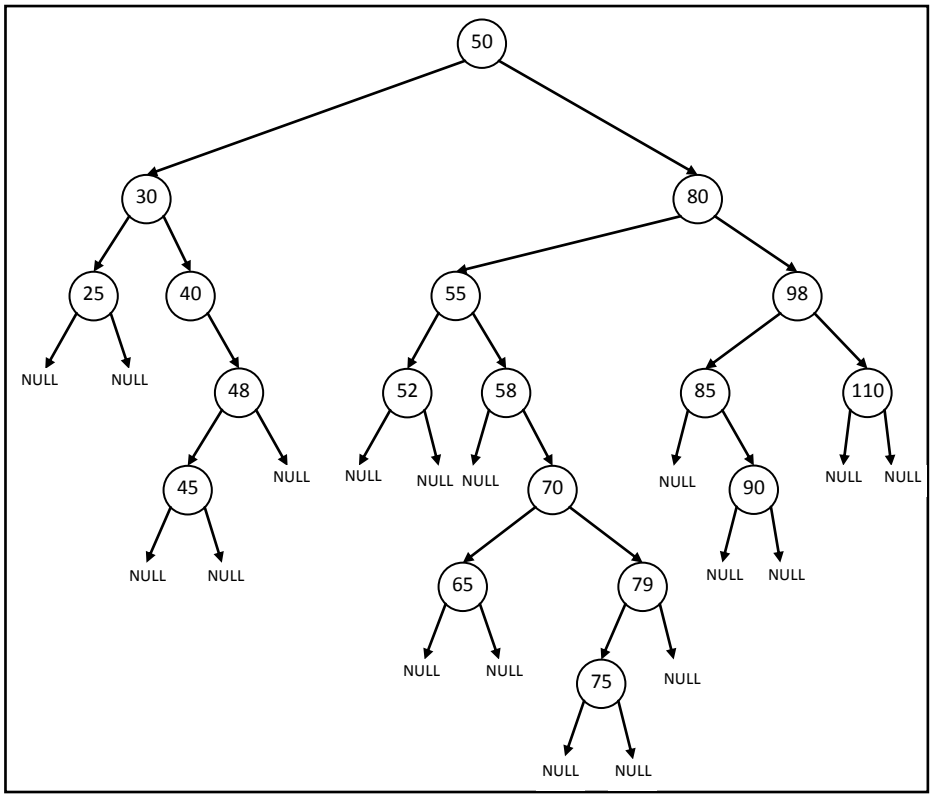
[10 marks]

6. The following code lists the nodes in a binary tree in two different orders:

```
Preorder: ABCDEFGHIJKLM
Inorder:  CEDFBAHJIKGML
```

Draw the binary tree.

[10 marks]



5. *Merujuk kepada pepohon gelintaran perduaan di atas,*
   (a) Senaraikan laluan dari nod berlabel 80 ke nod berlabel 79.
   (b) Suatu nod berlabel 35 akan diselitkan ke dalam pepohon. Lukis semula pepohon gelintaran perduaan selepas selitan.
   (c) Hapuskan nod berlabel 52 dan lukis semula pepohon gelintaran perduaan tersebut.
   (d) Hapuskan nod berlabel 40 dan lukis semula pepohon gelintaran perduaan tersebut.
   (e) Hapuskan nod berlabel 80 dan lukis semula pepohon gelintaran perduaan tersebut.

   [10 markah]

6. *Kod-kod yang di bawah senaraikan nod-nod sesuatu pepohon perduaan dalam dua tertib yang berbeza:*

```
Tertib awalan:   ABCDEFGHIJKLM
Tertib sisipan:  CEDFBAHJIKGML
```

   Lukiskan pepohon perduaan tersebut.

   [10 markah]

7. Consider the following L-systems:
Let the alphabet $A = \{F, +, -, x, y\}$, axiom $S = x$, and the below production rules:

$$F \rightarrow F$$
$$+ \rightarrow +$$
$$-- \rightarrow -$$
$$x \rightarrow -yF + xFx + Fy -$$
$$y \rightarrow +xF - yFy - Fx +$$

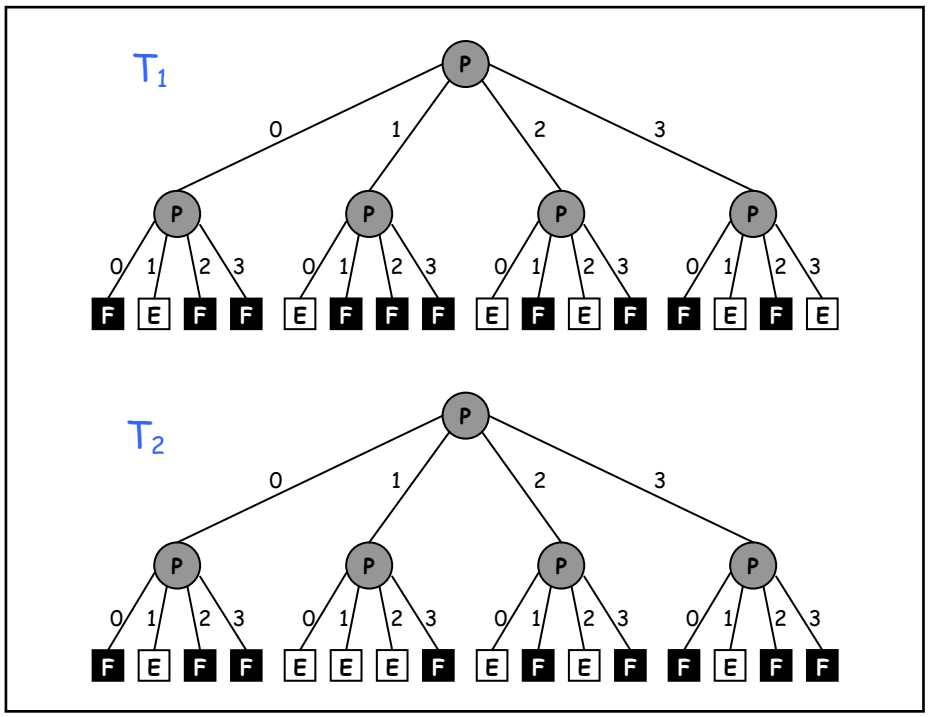
Here we give F, + and − these interpretations:

F means move one step forward while drawing a line.
\+ means turn counterclockwise by 90 degrees.
− means turn clockwise by 90 degrees.

(a) Assume that the axiom is at the first step, write the strings for the second and third steps, then
(b) draw the geometric realizations for the strings at second and third steps.

[10 marks]

8. Given two quadtrees, T_1 and T_2 as follows:



3	2
0	1

Figure 1

(a) Based on the above quadtrees, re-generate the original 4x4 pixel representations in 2-dimensional spaces, for T_1 and T_2. The order of the leaves numbers are given according to the sequence of the quadrants as in Figure 1.
(b) Find the quadtrees for $T_1 \cap T_2$ and $T_1 \cup T_2$.

[10 marks]

7. *Pertimbangkan system-L di bawah:*
Biar aksara A = {F, +, -, x, y}, aksiom S = x,dan peraturan-peraturan operasi di bawah:

$$F \rightarrow F$$
$$+ \rightarrow +$$
$$-- \rightarrow -$$
$$x \rightarrow -yF + xFx + Fy -$$
$$y \rightarrow +xF - yFy - Fx +$$

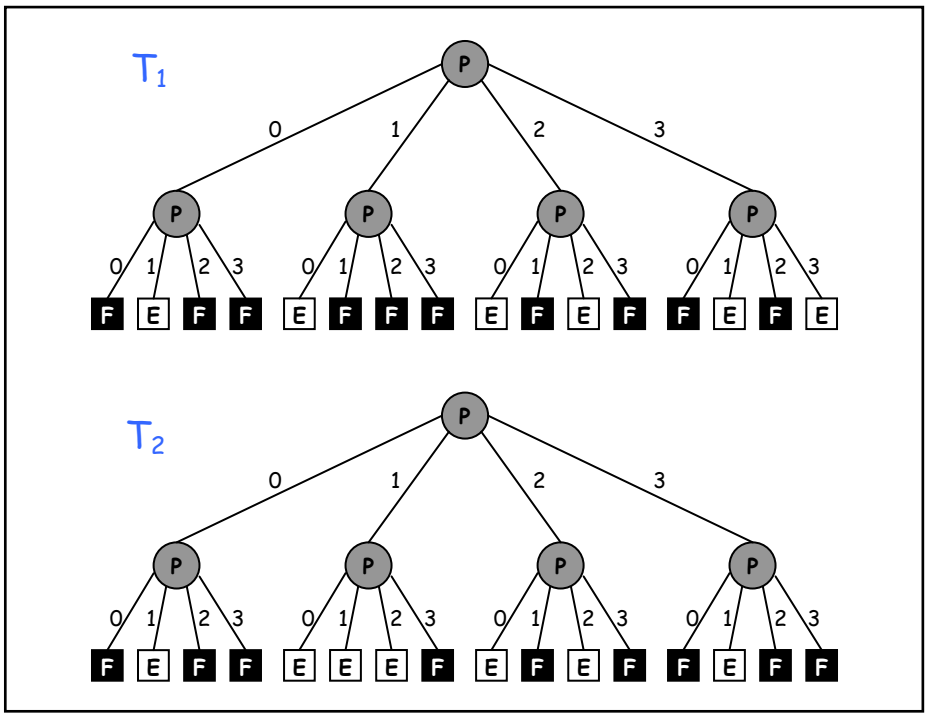
Disini, kita memberikan F, + dan – tafsiran di bawah:

F bermakna bergerak satu langkah ke hadapan semasa melukis satu garis.
+ bermakna putar arah songsang jam sebanyak 90 darjah.
– bermakna putar arah ikut jam sebanyak 90 darjah.

(a) Anggapkan aksiom tersebut berada pada langkah pertama, tuliskan rentetan-rentetan untuk langkah kedua and ketiga, kemudian
(b) lukiskan realisasi geometri-geometri untuk rentetan-rentetan tersebut pada langkah kedua dan ketiga.

[10 markah]

8. *Diberi dua pepohon kuad, T_1 dan T_2 seperti berikut:*



3	2
0	1

Gambarajah 1

(a) Berdasarkan kepada pepohon-pepohon kuad di atas, janakan semula perwakilan piksel 4x4 asal dalam ruang 2-dimensi, untuk T_1 and T_2. Nombor-nombor daun diberi mengikut tertib susunan kuardran seperti di Gambarajah 1.

(b) Cari pepohon-pepohon kuad untuk $T_1 \cap T_2$ dan $T_1 \cup T_2$.

[10 markah]

…10/-

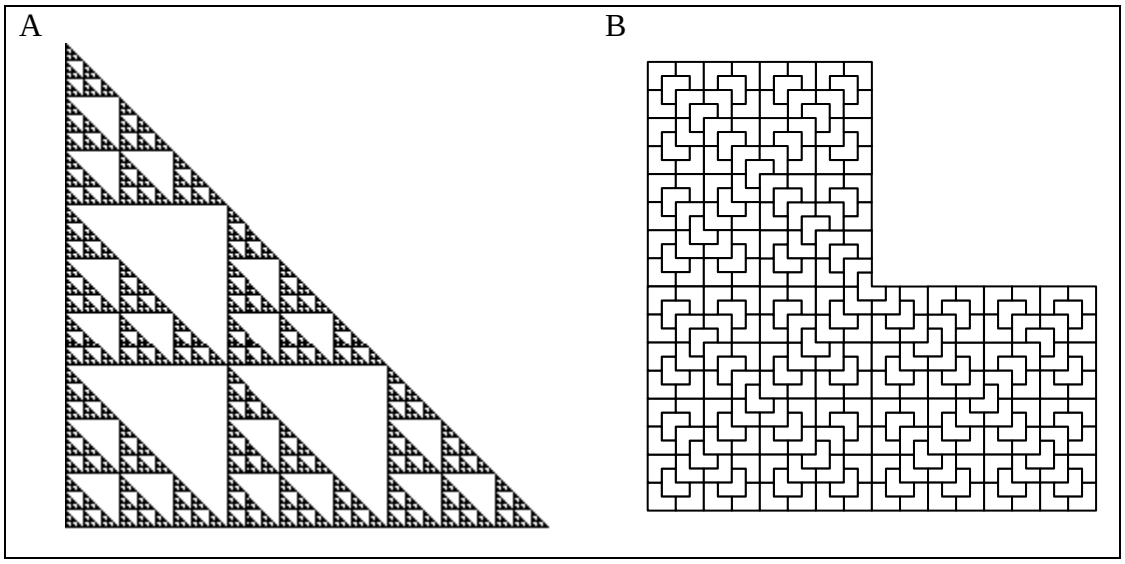


Figure 2

9. By using Box-Counting Dimension method, let r be the side length of a fractal and $N(r)$ be the number of boxes with side length r, which are needed to cover the shape. For each of the fractals A and B given in Figure 2, answer questions (a) to (c) as follows,

(a) find appropriate values of r and fill in the table below for $n = \{1, 2, 3, 4, 5\}$,

n	r_n	$N(r_n)$	—
:	:	:	:

(b) find a pattern in the values of $N(r_n)$, and

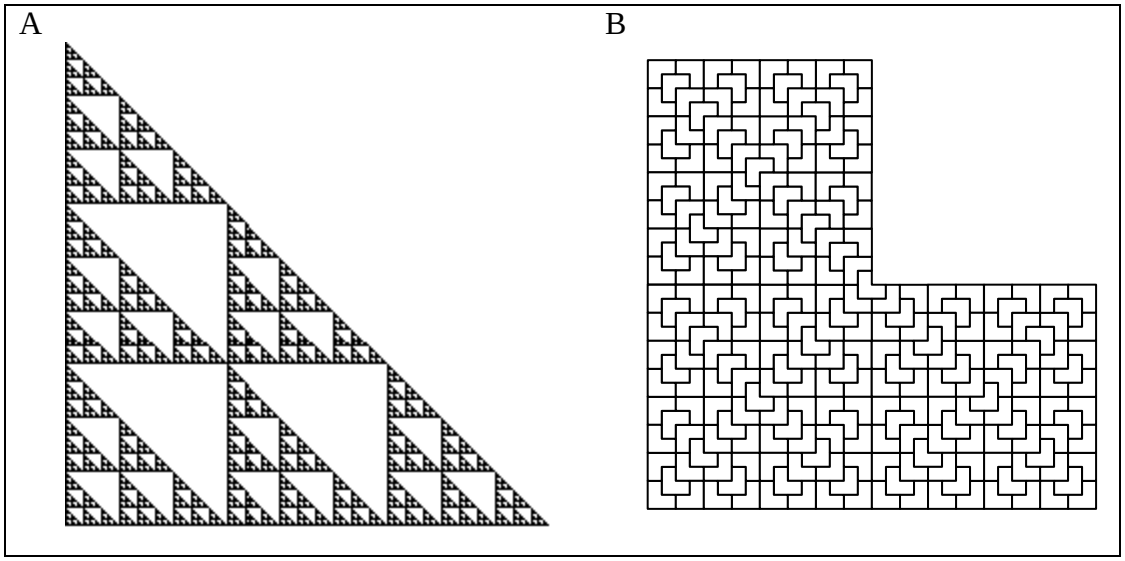
(c) use this pattern to compute the box-counting dimension, d_b, which is given as

$$d_b = \frac{\quad}{\quad}$$

$$= \frac{\quad}{\quad}$$

If the calculated dimension d_b is not an integer number, leave your answer in the simplest *Log* form, i.e. $d_b = \frac{\quad}{\quad}$.

[20 marks]



Gambarajah 2

9. *Dengan menggunakan kaedah Dimensi Pengiraan-Kotak, biar r sebagai panjang tepi untuk suatu fraktal dan N(r) sebagai bilangan kotak, dengan panjang tepi, r, yang dikehendaki untuk menutup bentuk fraktal tersebut. Untuk setiap fraktal A dan B yang ditunjuk di Gambarajah 2, jawab soalan-soalan dari (a) ke (c) seperti berikut,*

   (a) Cari nilai-nilai yang sesuai untuk r dan isikan jadual di bawah untuk n = {1, 2, 3, 4, 5}.

n	r_n	$N(r_n)$	—
:	:	:	:

   (b) Cari satu corak dalam nilai-nilai $N(r_n)$, dan

   (c) gunakan corak tersebut untuk mengira dimensi pengiraan-kotak, d_b, yang diberi sebagai

$$d_b = \frac{\quad}{\quad}$$

$$= \frac{\quad}{\quad}$$

Jika dimensi d_b yang terdapat bukan suatu nombor integer, biarkan jawapan anda dalam bentuk Log, i.e. $d_b = \frac{\quad}{\quad}$, yang paling mudah.

[20 markah]

- oooOooo -