UNIVERSITI SAINS MALAYSIA

Second Semester Examination
[Peperiksaan Semester Kedua]

Academic Session 2007/2008
[Sidang Akademik 2007/2008]

April 2008

CPT212/CPT201 – Design & Analysis of Algorithms
[Reka Bentuk & Analisis Algoritma]

Duration : 2 hours
[Masa : 2 jam]

INSTRUCTIONS TO CANDIDATE:
[ARAHAN KEPADA CALON:]

- Please ensure that this examination paper contains **FIVE** questions in **SEBELAS** printed pages before you begin the examination.

  *[Sila pastikan bahawa kertas peperiksaan ini mengandungi **LIMA** soalan di dalam **ELEVEN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer any **FOUR** questions only.

  *[Jawab mana-mana **EMPAT** soalan sahaja.]*

- You may answer the questions either in English or in Bahasa Malaysia.

  [Anda dibenarkan menjawab soalan sama ada dalam Bahasa Inggeris atau Bahasa Malaysia.]

---

1. (a) (i) An algorithm is required to insert an element x into an ordered array A of size n and the array is not full. The algorithm is to locate sequentially the correct position for insertion so that the array remains ordered. The algorithm then shifts all the elements to the right of the position to create a space for the new element. Write a function in C++/Java that performs this tasks.

(ii) Determine the time complexity of your function in Question 1(a)(i) above for the best and worst cases in terms of n. Justify your answers.

(iii) What would be the time complexity of the algorithm in Question 1(a)(i) above if the sequential process in locating the correct position is replaced by a binary search. Explain you answer.

(40/100)

(b) Given below is the selection-sort algorithm:

```
template<class T>
void selectionsort (data[], int n)
{
    for (int i=0,j,least; i<n-1;i++)
            for (int j=i+1, least=i; j<n; j++){
                  if (data[j] < data[least])
                        least = j;
                  swap (data[least], data[i]); //line 8
            }
}
```

(i) Explain by referring to the above code why the time complexity of the algorithm is the same for all cases (best, average, and worst).

(ii) Even though the time complexity is the same for all cases there can be some savings in the number of swaps by replacing the line 8 of the code with:

```
if (data[i] != data[least])
      swap (data[least], data[i]);
```

Explain the above statement by giving a situation where we can save the number of swaps most and a situation where the number of swaps is maximum.

(30/100)

(c) Trace quicksort algorithm as it sorts the following array:

34 12 45 41 23

(30/100)

2. (a) (i) Rank the worst case performances of the following operations on a binary search tree from the lowest to the highest (Group them together in a bracket if they are of the same complexity):

Searching, Traversal, Insertion, Deletion, and Treesort

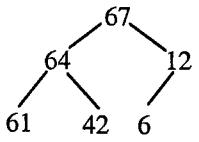
(ii) Redo the task in Quesion 2(a)(i) above for operations on an AVL tree.

(35/100)

(b) A student has two options of using linked lists in representing a priority queue namely by having all elements entry ordered, and another order by putting a new element in its proper position according to its priority. Advise the student on the option that the student should take by considering various scenarios based on the advantages and disadvantages of each representation.

(20/100)

(c) Write a pseudocode (C++/Java) that will output a heap element by element and level by level. For example the following heap and its array representation:



67	64	12	61	42	6	.....	

will be output as:

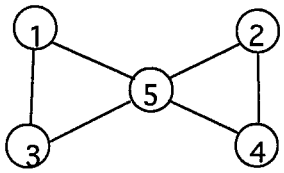
67
64 12
61 42 6

(30/100)

(d) Perform heapsort on the heap given in Question 2(c) above.

(15/100)

3. (a) Given the following graph:



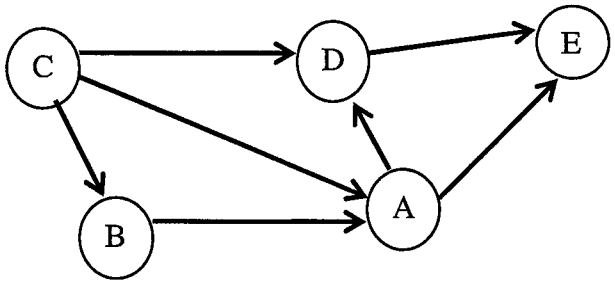
(i) Give the adjacency matrix representation of the graph.

(ii) Give the incidence matrix representation of the graph.

(iii) What are the advantages of the incidence matrix representation over the adjacency matrix representation in terms of space and time complexities?

(35/100)

(b) Show the steps involved in carrying out topological sort on the graph below using the minimal vertex approach (give only one of the possible outputs):



(20/100)

(c) A hash table uses a hash function hash(key) = (key+1) % 13 and the table size is 13.

(i) Is the above hash function a good hash function? Justify your answer.

(ii) Write a function in C++/Java for the above hash function.

(iii) Show the configuration of the hash table using the above hash function in Question 3(c)(i) above and linear probing technique to resolve collisions after the insertion of the keys 13, 16, 29, 15, 11, 42, 22, 31 into an initially empty table.

(45/100)

4. (a) Define the following terms in the context of processing data on secondary storage:

(i) Block

(ii) Read

(iii) Write

(25/100)

(b) Define a class in C++/Java for a node of a B-Tree.

(20/100)

(c) (i) Build a B-tree of order 5 by inserting the integers 13, 22, 19, 25, 24, 28, and 21 in the given order.

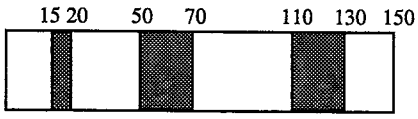
(ii) Show how searching process is carried out if key '10' and key '26' is searched in the above B-tree.

(35/100)

(d) Compare and contrast the 2-4 tree and the vh tree as a representation of a B-tree in terms of shape, flexibility and efficiency.

(20/100)

5. (a) Given the following heap:



(i) Give a series of memory allocations using the best-fit algorithm in order to show one of the disadvantages of the algorithm namely the remaining parts after returning the required size are small and practically unusable.

(ii) Give a series of memory allocations using the worst-fit algorithm in order to show one of the advantages of the algorithm namely the method prevents fragmentation by avoiding/delaying creation of small blocks.

(30/100)

(b) (i) Create a Huffman tree for five letters A, B, C, D, and E with probabilities 0.2, 0.3, 0.1, 0.25, and 0.15 respectively.

(ii) When would two different Huffman trees of the same average length for the same letters could be generated?

(30/100)

(c) Given below a pseudocode for a simple approach to string matching:

```
bruteForceStringMatching(pattern P, text T)
    i = 0;
    while i ≤ |T| - |P|
        j = 0;
        while Ti == Pj and j < |P|
            i++; // (A)
            j++;
        if j == |P|
            return match at i - |P|; // (B)
        i = i - j + 1; // (C)
    return no match;  // (D)
```

(i) Give appropriate comments to replace (A), (B), (C) and (D) as indicated in the above pseudocode.

(ii) The above algorithm finds an occurrence of a pattern in text and discontinue after finding the first. How could the algorithm be modified so that multiple searches could be carried out?

(40/100)

1. (a) (i) Sebuah algoritma diperlukan untuk menyisipkan sebuah unsur x ke dalam sebuah tatasusunan bertertib A bersaiz n dan tatasusunan tersebut tidak penuh. Algoritma berkenaan perlu melokasi secara berjujukan kedudukan yang betul untuk penyisipan supaya tatasusunan berkenaan tetap bertertib. Algoritma berkenaan kemudiannya menganjak semua unsur di sebelah kanan kedudukan berkenaan untuk mencipta satu ruang untuk unsur baru berkenaan. Tulis fungsi dalam C++/Java yang melakukan tugas tersebut.

(ii) Tentukan kekompleksan masa fungsi anda dalam Soalan 1(a)(i) di atas bagi kes terbaik dan terburuk dalam sebutan n. Justifikasikan jawapan anda.

(iii) Apakah kekompleksan algoritma dalam Soalan 1(a)(i) di atas jika proses berjujukan dalam melokasikan kedudukan yang betul diganti dengan gelintaran perduaan. Jelaskan jawapan anda.

(40/100)

(b) Diberi di bawah algoritma isihan pilih:

```
template<class T>
void selectionsort (data[], int n)
{
    for (int i=0,j,least; i<n-1;i++)
            for (int j=i+1, least=i; j<n; j++){
                  if (data[j] < data[least])
                        least = j;
                  swap (data[least], data[i]); //baris 8
            }
}
```

(i) Jelaskan dengan merujuk kepada kod di atas mengapa kekompleksan masa algoritma berkenaan adalah sama bagi semua kes (terbaik, purata, dan terburuk).

(ii) Walaupun kekompleksan masa adalah sama bagi semua kes, akan terdapat beberapa penjimatan bilangan penyilihan dengan menggantikan baris 8 kod berkenaan dengan:

```
if (data[i] != data[least])
      swap (data[least], data[i]);
```

Jelaskan kenyataan di atas dengan memberikan satu situasi apabila kita boleh menjimatkan paling banyak bilangan penyilihan dan satu situasi apabila bilangan penyilihan adalah maksimum.

(30/100)

(c) Surih algoritma isihan cepat apabila algoritma ini menyisih tatasusunan berikut:

34 12 45 41 23

(30/100)

2. (a) (i) Susun prestasi kes terburuk pengendalian-pengendalian berikut ke atas pepohon gelintaran perduaan daripada terendah kepada tertinggi (Kumpulkan bersama dalam sebuah kurungan jika sama kekompleksannya):

Penggelintaran, Penyusuran, Penyisipan, Penghapusan, dan Pepohon

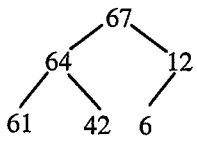
(ii) Lakukan semula tugas dalam Soalan 2(a)(i) di atas bagi pengendalian ke atas pepohon AVL.

(35/100)

(b) Seorang pelajar mempunyai dua pilihan untuk menggunakan senarai berpaut dalam mewakilkan baris gilir keutamaan iaitu dengan mempunyai semua unsur bertertib ikut kemasukan, dan satu lagi tertib yang meletakkan unsur baru pada kedudukan sepatutnya berdasarkan keutamaan. Beri nasihat kepada pelajar ini tentang pilihan yang sepatutnya diambil dengan mempertimbangkan pelbagai senario berasaskan kebaikan dan keburukan setiap perwakilan.

(20/100)

(c) Tulis pseudokod (C++/Java) yang akan memaparkan unsur demi unsur dan paras demi paras sebuah timbunan. Sebagai contoh timbunan berikut dan perwakilan tatasusunannya:



67	64	12	61	42	6	.....	

akan dioutput sebagai:

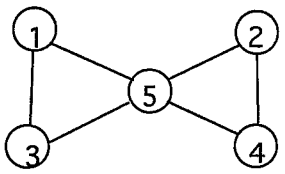
67
64 12
61 42 6

(30/100)

(d) Lakukan isihan timbun bagi timbunan yang diberikan dalam Soalan 2(c) di atas.

(15/100)

3. (a) Diberikan graf berikut:



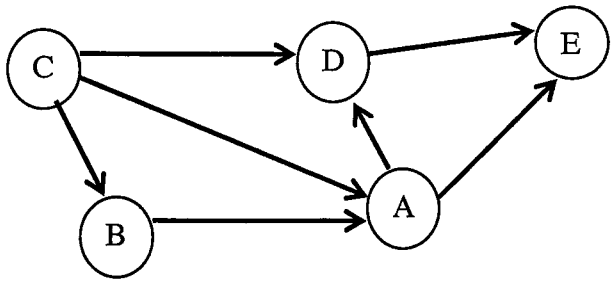
(i) Beri perwakilan matriks kesebelahan graf berkenaan.

(ii) Beri perwakilan matriks insidens graf berkenaan.

(iii) Apakah kebaikan perwakilan matriks insidens ke atas perwakilan matriks kesebelahan dari segi kekompleksan ruang dan masa?

(35/100)

(b) Tunjukkan langkah-langkah yang terlibat dalam menjalankan isihan topologi ke atas graf di bawah menggunakan pendekatan bucu minimum (beri hanya satu output yang mungkin):



(20/100)

(c) Sebuah jadual cincangan menggunakan fungsi cincangan hash(key) = (key+1) % 13 dan saiz jadual adalah 13.

(i) Adakah fungsi cincangan di atas sebuah fungsi cincangan yang baik? Justifikasikan jawapan anda.

(ii) Tulis fungsi dalam C++/Java bagi fungsi cincangan di atas.

(iii) Tunjuk konfigurasi jadual cincangan yang menggunakan fungsi cincangan dalam Soalan 3(c)(i) di atas dan teknik pencarian linear untuk meleraikan perlanggaran selepas penyisipan kunci-kunci 13, 16, 29, 15, 11, 42, 22, 31 ke dalam sebuah jadual yang asalnya kosong.

(45/100)

4. (a) Takrifkan istilah-istilah berikut dalam konteks pemprosesan ke atas storan sekunder:

(i) Blok

(ii) Baca

(iii) Tulis

(25/100)

(b) Takrifkan kelas dalam C++/Java bagi sebuah nod pepohon B.

(20/100)

(c) (i) Bina pepohon B bertertib 5 dengan menyisipkan 13, 22, 19, 25, 24, 28, dan 21 dalam tertib yang diberikan.

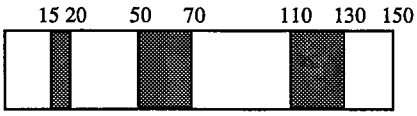
(ii) Tunjukkan bagaimana proses pengglintaran dilakukan jika kunci '10' dan kunci '26' digelintar dalam pepohon B di atas.

(35/100)

(d) Banding dan bezakan pepohon 2-4 dan pepohon vh sebagai perwakilan pepohon B dari segi rupa bentuk, keluwesan dan kecekapan.

(20/100)

5. (a) Diberi timbunan berikut:



(i) Beri satu siri peruntukan ingatan menggunakan algoritma penyuaian terbaik supaya satu daripada keburukan algoritma berkenaan ditunjukkan iaitu bahagian yang tinggal selepas memulangan saiz yang diperlukan adalah kecil dan tidak boleh digunakan secara praktik.

(ii) Beri satu siri peruntukan ingatan menggunakan algoritma penyuaian terburuk supaya satu daripada kebaikan algoritma berkenaan ditunjukkan iaitu menghalang penyerpihan dengan mengelak/melengahkan penciptaan blok-blok kecil.

(30/100)

(b) (i) Cipta pepohon Huffman bentuk lima huruf A, B, C, D, dan E dengan kebarangkalian masing-masing 0.2, 0.3, 0.1, 0.25, dan 0.15.

(ii) Bilakah dua pepohon Huffman yang berbeza mempunyai panjang purata yang sama bagi huruf-huruf yang sama boleh dihasilkan?

(30/100)

(c) Diberikan di bawah pseudokod untuk pendekatan mudah pemadanan rentetan:

```
bruteForceStringMatching(pattern P, text T)
    i = 0;
    while i ≤ |T| - |P|
        j = 0;
        while Ti == Pj and j < |P|
            i++; // (A)
            j++;
        if j == |P|
            return match at i - |P|; // (B)
        i = i - j + 1; // (C)
    return no match;  // (D)
```

(i) Beri ulasan yang sepatutnya bagi menggantikan (A), (B), (C) dan (D) seperti yang ditandakan dalam pseudokod di atas.

(ii) Algoritma di atas mencari satu kejadian sebuah pola dalam teks dan memberhentikannya apabila yang pertama ditemui. Bagaimanakah algoritma berkenaan diubahsuai supaya penggelintaran berganda boleh dilakukan?

(40/100)

-oooOoooo-