UNIVERSITI SAINS MALAYSIA

Stamford College

First Semester Examination
2002/2003 Academic Session
September 2002

External Degree Programme
Bachelor of Computer Science (Hons.)

CPT201/CTP201 – Design & Analysis of Algorithms

Duration : 3 hours

INSTRUCTIONS TO CANDIDATE:

- Please ensure that this examination paper contains **FIVE** questions in **SIX** printed pages before you start the examination.
- Answer **FOUR** questions only.
- On each page, write *only your **Student ID***.

1. (a) (i) Arrange the following expressions by growth rate from slowest to fastest.

$$n \quad \log_2 n \quad n^2 \quad n\log_2 n \quad n^3 \quad 2^n \quad 1$$

(ii) Explain why the concept of growth rate is extremely important.

(15/100)

(b) Consider the problem of finding a value with the largest number of occurrences in a sorted array.

(i) Describe a possible solution to this algorithm, and write pseudocodes for the algorithm.

(ii) What will be the complexity/efficiencies of this algorithm in O-Notation?

(25/100)

(c) Almost all of the sorting algorithms that you have studied operate on an array. What would be the effect on the following algorithms if they are made to operate on a linked list? For each of the algorithms, describe a possible solution (do not write any codes), problems encountered (if any) and whether the linked-list implementation would affect the complexity/efficiency of the algorithm.

(i) Insertion sort.

(ii) Mergesort.

(iii) Quicksort.

(30/100)

(d) (i) The function `quicksort` (as given in lecture) uses the function `choosePivot` (as given in lecture) to choose a pivot and place it into the first array location. Describe two ways of implementing `choosePivot` that you can think of.

(ii) Perform `quicksort` on the following integers using each of the two ways that you have come up with in 1(d)(i) above.

26 38 10 35 26 7

(30/100)

2. (a) The basic idea of the treesort algorithm is as follows (as given in lecture).

```
Treesort (inout anArray: ArrayType, in n: integer)
// Sorts the n integers in an array anArray into ascending order.

Insert anArray's elements into a binary search tree bTree.

Traverse bTree in inorder. As you visit bTree's nodes, copy their
    data items into successive locations of anArray.
```

(i) Using your own words describe in detail the above algorithm.

(ii) How would you modify the above algorithm so that we can sort an array into descending order instead of into ascending order?

(25/100)

(b) The non-linear, binary search tree implementation of the ADT table is a better choice in general over the two linear implementations. Give the advantages and disadvantages of the binary search tree implementation for the ADT table, paying special attention to its merits over the two linear implementations.

(25/100)

(c) In our definition of a heap, the root contains the item with the largest search key. Such a heap is known as a maxheap. A minheap, on the other hand, places the item with the smallest search key in its root.

(i) Rewrite the definition of the heap based on the minheap.

(ii) Where in a min-heap might the largest element reside?

(iii) Show the minheap that results from by inserting the following integers into the heap in the given order.

10 5 12 8

(iv) What would happen if the function `heapsort` (as given in lecture) for a maxheap, is applied to a minheap (without any modification to the function)? Explain.

(v) Assuming the function `heapRebuild` (as given in lecture) has been modified for minheap, perform heapsort (into a descending order) on the minheap obtained in 2(c)(iii) above.

(50/100)

3. (a) Give the complexity/efficiency of searching for each of the following search structures.

(i) 2-3 tree.

(ii) 2-3-4 tree.

(iii) Red-black tree.

(iv) AVL tree.

(20/100)

(b) If your application of the ADT table involves only retrieval such as a thesaurus – what tree from (a) above would provide the most efficient table implementation? Justify your answer.

(15/100)

(c) Describe the basic strategy of the AVL method in balancing binary search tree. Your description should include the various types of rotation involved together with some graphical illustrations.

(25/100)

(d) Assume that you are hashing key k to a hash table of n slots (indexed from 0 to n-1). For each of the following functions h(k), is the function acceptable as a hash function (i.e. would the hash function work correctly for both insertions and searches), and if so, is it a good hash function? Justify your answer.

(i) `h(k) = k/n` where k and n are integers.

(ii) `h(k) = 1`.

(iii) `h(k) = (k + Random(n)) mod n`, where function `Random(n)` returns a random integer between 0 and n-1, inclusive.

(iv) `h(k) = k mod n`, where n is a prime number.

(20/100)

(e) One way of hashing a string of characters to an integer is by firstly summing up the ASCII values of each characters and finally applying the modulo operation.

(i) Write the hash function in C++.

(ii) Is it a good hash function? Justify your answer.

(20/100)

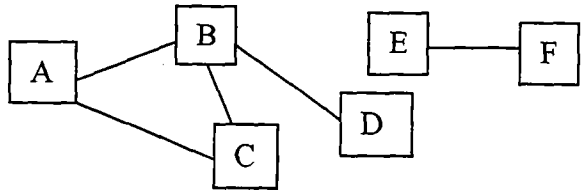
4. (a) An example of an application of graphs is navigation of a city graph. The user of such an application may start the navigation at any one of the cities and then go to one of the cities reachable from the current city. The navigation then resume with the chosen city. The navigation is then repeated as before until the user quits. Discuss suitability of using adjacency matrix representation, and adjacency list representation for such an application.

(20/100)

(b) (i) Under what condition does the graph traversal algorithm visit only a subset of the graph's vertices?

(ii) This subset is called the connected components containing v where v is the starting vertex of the traversal. How do you determine all the connected components of a given graph?

(iii) Determine all the connected components of the following graph:



(iv) Write C++ pseudocode to determine the number of connected components in a given graph. You may use the dfs or bfs functions (as given in lecture).

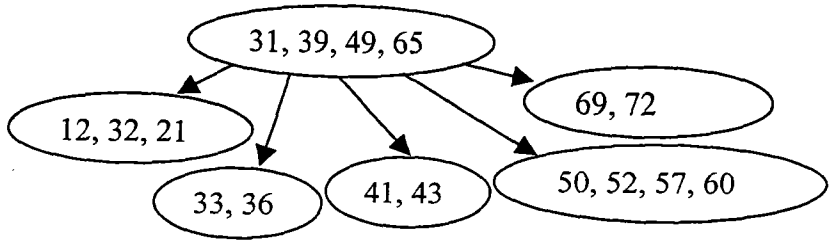
(35/100)

(c) An external file contains 12 records. You want to sort these records. The internal memory of the computer can only support 3 records at one time. Illustrate how the external mergesort sorts the records. The records are as follows: S O A L A N M U D A H A.

(15/100)

(d) (i) Describe the algorithm for `tableRetrieve` (as given in lecture) for an external B-Tree (Please do not write any codes/pseudocodes).

(ii) Given below is a B-tree of degree 5. Draw the B-Tree that results from deleting 21 from the tree.



(30/100)

5. (a) (i) Describe the first-fit method and the best-fit method.

(ii) Which of these strategies (methods) is best?

(25/100)

(b) (i) Describe the format for a freelist, a freeblock and what happen to a free block before and after allocation. Include in your answer some graphical illustrations and how to obtain the address of the new allocated block.

(ii) Why would a doubly linked-list implementation be more superior to an ordinary linked-list implementation for freelists?

(30/100)

(c) Given the following algorithm that finds a substring in a given string.

```
int Find (const String& source, const String& target)
// Return the position in source of the first occurrence of
// substring target, and returns -1 if unsuccessful
{  if (source or target is empty)
      return -1;          // no match possible
   int current = 0;       // possible location in this
   while (complete match not found
                          && any characters left in source)
      if (current character in source != first target
                                                 character)
            current++;
      else {do                    // found a partial match
                  Step through source and target together
            while (chars left to compare && still
                                              have a match);
            if (no more target characters to inspect)
                  return current;  // found a full match.
            else current++; // keep looking
      }
   return -1; // no match found
}
```

Beside theoretical analysis, we may also perform empirical analysis. Modify the above algorithm to include some measurements as needed, so that we may carry out empirical analysis on it.

(20/100)

(d) (i) Define exhaustive search.

(ii) Briefly illustrate your answer to 5(d)(i) above with a problem that can be solved using exhaustive search.

(25/100)

- oooOooo -