Matriculation Number:_____ **CS1102C**

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER I (2005-2006)

EXAMINATION FOR
CS1102C: DATA STRUCTURES AND ALGORITHMS

November 2005
Time Allowed: 2 Hours

MATRICULATION NUMBER:

INSTRUCTIONS TO CANDIDATES:

1. Write your matriculation number in the space provided above. **Also write your matriculation number at the top of each sheet in the exam paper.**
2. This examination paper consists of **10 questions** and comprises **14** printed pages including this front page.
3. Answer all of the questions directly in the space given after each question. If necessary, use the back of the page.
4. Marks allocated to each question are indicated. Total marks for the paper is **100.**
5. This is a closed book examination and you can write in pencil.

EXAMINER'S USE ONLY				
Questions	**Possible**	**Marks**	**Grader**	**Check**
Q1	21			
Q2	10			
Q3	12			
Q4	5			
Q5	10			
Q6	10			
Q7	8			
Q8	10			
Q9	4			
Q10	10			
Total	100			

Question 1 (21 marks)

Present **in pseudo code** correct and efficient algorithms to convert between the following graph data structures, for an undirected graph G with n vertices and m edges. You must give the **time complexity** of each algorithm.

(a) Convert from an **adjacency matrix** to an **adjacency list**.

Question 1(b)

(b) Convert from an **adjacency list** to an **incidence matrix**. An incidence matrix M has a row for each vertex and a column for each edge, such that $M[i,j]=1$ if vertex i is part of edge j, otherwise $M[i,j] = 0$.

Question 1(c)

(c) Convert from an **incidence matrix** to an **adjacency list**.

Question 2 (10 marks)

For each of the questions below, **briefly explain** your answer.

(a) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ on some inputs?

(b) If I prove that an algorithm takes $O(n^2)$ worst-case time, is it possible that it takes $O(n)$ on all inputs?

(c) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ on some inputs?

(d) If I prove that an algorithm takes $\Theta(n^2)$ worst-case time, is it possible that it takes $O(n)$ on all inputs?

(e) Is $f(n) = \Theta(n^2)$, where $f(n) = 100n^2$ for even n and $f(n) = 20n^2 - n \log n$ for odd n?

Question 3 (12 marks)

Use the hash function h(x) = x % 11 to map an integer value to a hash table index.
Insert the data 1, 13, 12, 53, 77, 29, 31, 22 into the hash table.
(a) Construct the hash table by using linear probing.

(b) Construct the hash table by using separate chaining.

(c) Determine the load factors for **both** techniques.

Question 4 (5 marks)

Write in C++ a **recursive** function that will print a given positive integer vertically. For example, 1234 is printed as

1
2
3
4

Question 5 (10 marks)

Given an unsorted array of int, describe an **efficient** function to list all numbers in the array with values within the range [x..y]

(a) When the result is not required to be sorted

(b) When the result should be sorted.

Question 6 (10 marks)

STL queue supports the following functions:

back	returns a reference to last element of a queue
empty	true if the queue has no elements
front	returns a reference to the first element of a queue
pop	removes the top element of a queue
push	adds an element to the end of the queue
size	returns the number of items in the queue

Use the functions in STL queue to implement the following functions **in C++**:

(a) void split (queue<int>& q1, queue<int>& q2)

Split a queue q1 in two with q1 contains the front half, and q2 the remaining elements. If there are 7 elements, then q1 will contain 3 elements and q2 will contain 4 elements.

Question 6(b)

(b)void combine(queue<int>& q1, const queue<int>& q2)

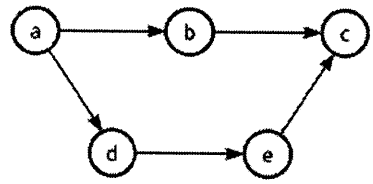
Combine two queues q1 and q2 by placing all elements of q2 at the end of q1 such that the relative order of elements from the second queue is unchanged. **The contents of q2 should remain unchanged**.

Question 7 (8 marks)

Assume that the queue is implemented in an int array a[]with ***theFront*** and ***theBack*** as the indices of the front and the back of the queue, and ***theSize*** as the size of the array. When the queue is full, the array has to be extended. Implement **in pseudo code** to double the size of the array when it is full.

Question 8 (10 marks)

What is the data structure used in the topological sorting algorithm **toposort**? Given the following graph, trace the **toposort** algorithm by showing the details about vertices being moved around and printed from a data structure.



Question 9 (4 marks)

Mr. B. C. Dull claims to have developed a new data structure for priority queues that supports the operations *Insert* and *Extract-Max*, all in $O(1)$ worst-case time. Show Mr. Dull that it is impossible.

Question 10 (10 marks)

Suppose you have access to a balanced binary search tree, which supports each of the operations search, insert, delete, minimum, maximum, successor, and predecessor in O(log n) time. Explain how to modify the insert and delete operations so that they still take O(log n) but now minimum and maximum take $O(1)$ time.

END-OF-PAPER