The University of Melbourne
Department of Computing and Information Systems

COMP20007

Design of Algorithms
June Assessment, 2013

Student Number:

Identical Examination papers: None.

Common Content: None.

Exam Duration: Three hours.

Reading Time: Fifteen minutes.

Length: This paper has 15 pages including this cover page.

Total Marks: 85

Authorized Materials: None.

Instructions to Invigilators: Students will write all of their answers on this examination paper. Students may not remove any part of the examination paper from the examination room.

Instructions to Students: This paper counts for 60% of your final grade. All questions should be answered in the spaces provided on the examination paper. You may make rough notes, and prepare draft answers, on the reverse of any page, and then copy them neatly into the boxes provided. You are not required to write comments in any of your code fragments or functions.

Throughout you should assume a RAM model of computation where input items fit in a word of memory, and basic operations such as $+ - \times /$ and memory access are all constant time.

Calculators: Calculators are not permitted.

Library: This paper may not be held by the Baillieu Library.

Question 1 (18 marks).

(a) (2 marks) Define an Euler Path in a directed graph $G(V, E)$ that starts at vertex $u \in V$.

(b) (4 marks) Draw the de Bruijn graph for text fragments of length $k = 3$ that would be formed from the text fragments $\{CGA, GAG, AGT, GTG, TGA\}$. You should not include vertices that have no incident edges.

(c) (2 marks) Complete the following C code function.

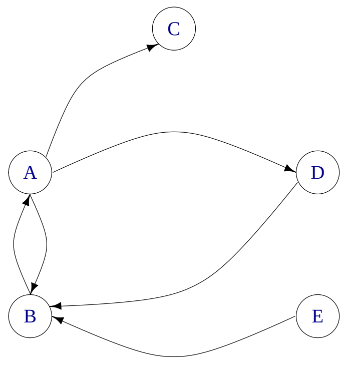
```
#define N 100
/*
** INPUTS: G is an N by N adjacency matrix of a graph
**         where G[u][v]=1 indicates an edge from u to v.
** RETURNS: the vertex number of any vertex that has an out degree
**          greater than zero.
** SIDE EFFECTS: None.
*/
int
find_v(int G[N][N]) {



}
```

(d) (2 marks) Assuming that there is an Euler path from vertex v back to vertex v that uses all edges in a graph G, what can you say about the degrees of every vertex in the graph?

(e) (6 marks) Outline an efficient algorithm for finding a Euler path in a directed graph G. You should assume that it is known that an Euler path does exist for G, just like in the graphs for Assignment 2. You do not need to give any implementation details or data structures. I am expecting less than 10 lines of high-level pseudo code.

(f) (2 marks) What is the time complexity of your algorithm in Part (e)? Give a very brief, high-level justification; again, no detail on data structures or implementation is required.

Question 2 (9 marks).



(a) (5 marks) For the above graph, complete the table of pre and post numbers for each vertex when a Depth First Search is performed beginning at A. If there is a choice of edge, choose the lower letter vertex in lexicographic order. Begin your numbering at 1.

Vertex	Pre number	Post number
A		
B		
C		
D		
E		

(b) (1 mark) List any back edges discovered in the DFS from (a).

(c) (1 mark) List any cross edges discovered in the DFS from (a).

(d) (1 mark) Name one source vertex in the graph at the top of this page.

(e) (1 mark) Name one sink vertex in the graph at the top of this page.

Question 3 (7 marks).

(a) (4 marks) Show the Huffman tree for the input frequencies $\{1, 1, 1, 3, 8, 5, 5, 26\}$.

(b) (1 mark) If the input frequencies in (a) were the actual frequency counts of symbols in a message to be compressed with the Huffman code, how many total bits would the symbol with frequency 8 require in the compressed output, ignoring the cost of storing a description of the code?

(c) (2 marks) Write down an expression for Shannon's entropy of probability distribution $p = \{p_1, p_2, \ldots, p_n\}$.

Question 4 (10 marks).

(a) (5 marks) What is the Burrows Wheeler Transform of the string $abcabc\$$? Assume that \$ is less than all other characters.

(b) (5 marks) Draw the Wavelet Tree for your BWT string in (a) assuming the shape of the tree is based on the prefix code: \$=`00`, a=`01`, b=`10`, c=`11`
You should show both the bitvector and the character strings at each node, even though only the bitvectors are actually stored.

Question 5 (12 marks).

(a) (3 marks) What are the attributes of a problem that would indicate that applying Dynamic Programming might lead to an efficient (polynomial) algorithm to solve the problem?

(b) (6 marks) Given two strings $x[1..n]$ and $y[1..m]$, the Edit Distance between the two strings can be calculated as $E(n, m)$, where
$E(0, j) = E(i, 0) = 0$;
$\delta(i, j) = 1$ when $x[i] \neq y[j]$ and 0 otherwise; and
$E(i, j) = \min\{1 + E(i-1, j), 1 + E(i, j-1), \delta(i, j) + E(i-1, j-1)\}$ when $i \in [1, n]$ and $j \in [1, m]$.

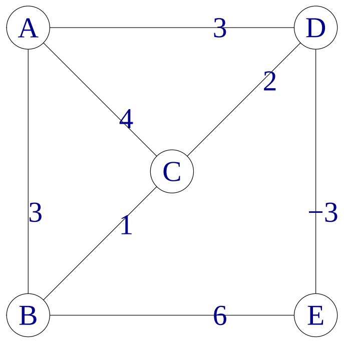
Using the Dynamic Programming technique, give pseudo code for a polynomial time algorithm for computing $E(n, m)$.

(c) (1 mark) What is a tight upper bound on the worst case running time of your algorithm in (c)?

(d) (1 mark) What is a tight lower bound on the best case running time of your algorithm in (c)?

(e) (1 mark) Given your answers to (d) and (e), is there a precise statement you can make about the running time of your algorithm that holds for best, average and worst case inputs? If so, what is it?

Question 6 (6 marks).



(a) (4 marks) Draw two different minimum spanning trees for the graph above.

(b) (2 marks) What are the first two edges added to the partial solution by Kruskal's greedy algorithm for building a Minimum Spanning Tree on the above graph?

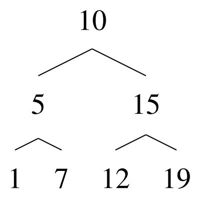
Question 7 (9 marks).

Answer True (T) of False (F) to the following statements. You will **lose** one mark for an incorrect answer in this question. Do not guess. The minimum possible mark for this question is zero.

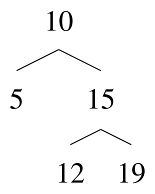
$f(n) = n^3$ is in $\Omega(n)$	
$f(n) = n^3 \log n$ is in $O(n^3)$	
$f(n) = n^2/\log n$ is in $O(n^2)$	
$f(n) = \log\log n$ is in $O(\log^* n)$	
$f(n) = \sqrt{n}$ is in $\Omega(\log n)$	
It is possible to devise an algorithm that can answer a `rank` query on a bitvector in $O(1)$ time using no more than $2n$ extra bits of space	
It is possible to devise an algorithm that can sort an array of integers in $O(n)$ time if the maximum integer is restricted to be $O(n)$.	
The travelling salesman problem is in class NP, but the minimum spanning tree problem is in P.	
Algorithms are still fun (no -1 for incorrect on this one; Alistair suggests -3 for incorrect!).	

Question 8 (14 marks).

(a) (2 marks) Draw this tree after rotating the root once to the right so that 5 becomes the new root.



(b) (5 marks) Draw this AVL tree after inserting 13. Show the tree before rebalancing and after each rotation.



After inserting 13

After first rotation

After second rotation

(c) (7 marks) Complete the following table for each of these data structures using big-Oh expressions assuming they contain n items.

What is the space required for a hash table that uses Open Addressing and has m slots?	
What is the space required for an AVL tree?	
What is the worst case time to find an item in an AVL tree?	
What is the space required for a Wavelet Tree based on a prefix code with all codewords of $\lceil \log_2 n \rceil$ bits?	
What is the worst case time required to find the successor of an item in an AVL tree?	
What is the worst case time required to find the successor of an item in a Hash Table that has m slots and uses Open Addressing for collision resolution?	
What is the worst case time required to find an item in a Hash Table that uses separate chaining and has a load factor of n?	

Overflow Answers

The boxes here are for emergency use only. If you do need to use this page, indicate **CLEARLY** in your previous answer that you have continued onto this page. Without such an indication, it is possible that this part of your answer will be overlooked.