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 $u \in V$.

ANSWER

1 marks A path in G that visits every edge of G precisely once

1 mark and is also a cycle from u to u.

(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.

ANSWER

2 marks 5 vertices

2 marks the edges.

Turn Over

(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]) {
```

ANSWER
```
0.5 mark   for (i = 0 ; i < N ; i++)
0.5 mark   for (j = 0 ; j < N ; j++)
0.5 mark   if (G[i][j] == 1)
0.5 mark       return i; // no marks for return j
```

So one mark for some double loop
and 1 mark for getting the right thing

```
}
```

(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?

ANSWER
2 marks The in-degree of every vertex must equal its out-degree.

(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.

ANSWER

0.5 mark	Choose some vertex in G, say v.
0.5 mark	Set the path P to a cycle from v back to v.
0.5 mark	While P does not contain all edges in G
0.5 mark	Find a vertex in P that has an unvisited edge leading out, say u
0.5 mark	Let q be a cycle from u back to u
0.5 mark	Replace u in P with q
0.0 mark	Return P

This is straight out of Assignment 2 spec.
Who knows what the students will serve up!

(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.
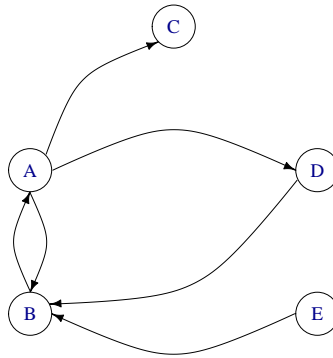
ANSWER

1 mark	$O(m)$ or $O(E)$
1 mark	Constant amount of work per edge.

OR (not really correct, but close enough)

1 mark	$O(m^2)$ or $O(E^2)$
1 mark	Loop goes around $O(m)$ times and "find vertex" could be $O(m)$

If they write a silly alg in (e), they can still get the marks here.

     Turn Over

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.

ANSWER

Vertex	Pre number	Post number
A	1	8
B	2	3
C	4	5
D	6	7
E	9	10

Half a mark per cell

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

ANSWER

  1 mark   BA

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

ANSWER

  1 mark   DB

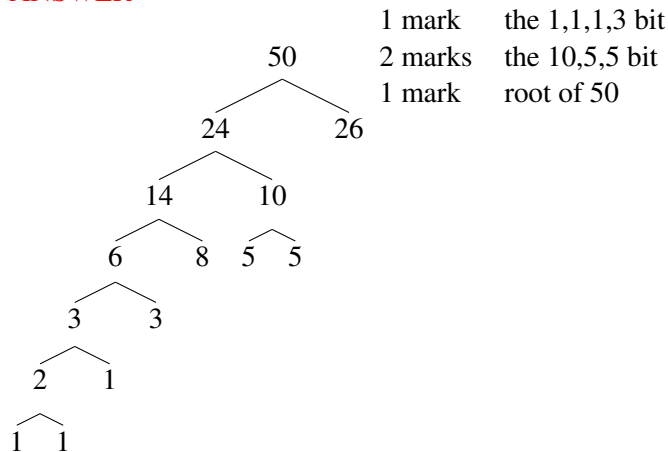(d) (1 mark) Name one source vertex in the graph at the top of this page.

ANSWER

  1 mark   E

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

ANSWER

  1 mark   C

        Turn Over

Question 3 (7 marks).

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

ANSWER

$$
\begin{array}{l}
50 \\
2426 \\
1410 \\
6855 \\
33 \\
21 \\
11
\end{array}
$$

1 mark	the 1,1,1,3 bit
2 marks	the 10,5,5 bit
1 mark	root of 50

(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?

ANSWER
$8 \times 3 = 24$

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

ANSWER

$- \sum_{i=1}^{n} p_i \log_2 p_i.$

0.5 mark	negative out the front
0.5 mark	some sum over all symbols
0.5 mark	some log base 2
0.5 mark	p_i's in the right spot

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.

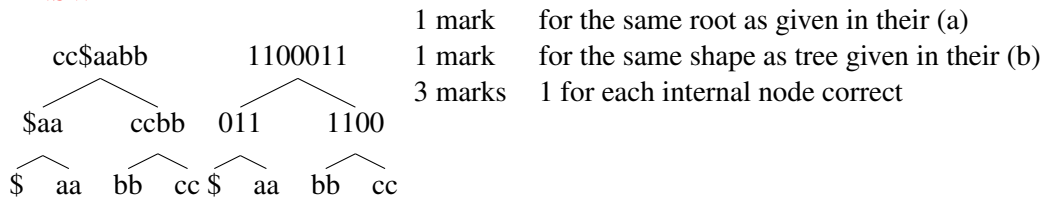ANSWER
$abcabc
abc$abc
abcabc$
bc$abca
bcabc$a
c$abcab
cabc$ab
(3 mark) for attempting to sort suffixes/rotations
(2 mark) BWT = cc$aabb

(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.

ANSWER

1 mark	for the same root as given in their (a)
cc$aabb 1100011	1 mark	for the same shape as tree given in their (b)
	3 marks	1 for each internal node correct

$aa   ccbb   011   1100

$ aa   bb   cc $ aa   bb   cc

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?

ANSWER

0.5 marks	Problem can be decomposed into sub-problems
0.5 marks	The solution can be composed of the solutions to the sub-problems
0.5 marks	The sub-problems overlap (repetition)
0.5 marks	The sub-problems can be dependency ordered
1 marks	There is a polynomial number of sub-problems

(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 $E(n, m)$.

ANSWER

1 marks	Allocate a two dimensional table[n][m]
1 marks	Set $t[0][*] = 0$ and $t[*][0] = 0$
0.5 marks	for i in 1 to n
0.5 marks	for j in 1 to m
0.5 marks	$\quad a = 1 + t[i - 1][j]$
0.5 marks	$\quad b = 1 + t[i][j - 1]$
1 marks	$\quad c = (x[i] \neq y[j]) + t[i - 1][j - 1]$
0.5 marks	$\quad t[i][j] = \min(a, b, c)$
0.5 marks	return $t[n][m]$

2 marks for mentioning table and initialise
1 mark for double loop over table
1 mark for the two easy cases
1 mark for the harder case
1/2 mark for min of three cases
1/2 mark for saying $t[n][m]$ is answer

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

ANSWER

1 marks $O(nm)$

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

ANSWER

1 marks $O(nm)$

(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?
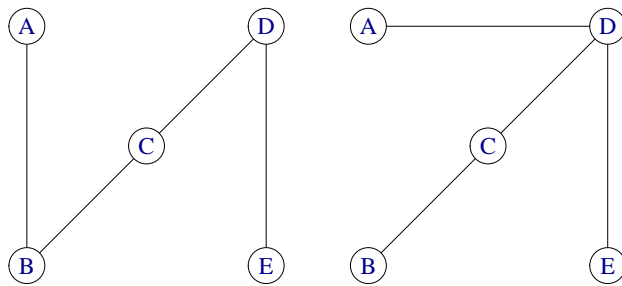
ANSWER

1 marks $\Theta(nm)$

Award the mark if wrong but consistent with (d) and (e).

Question 6 (6 marks).
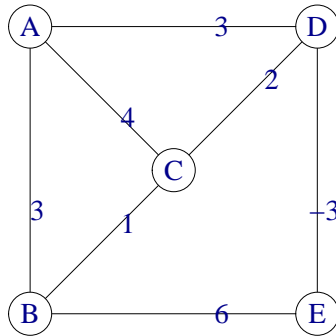
(a) (4 marks) Draw two different minimum spanning trees for the graph above.

ANSWER

2 marks for each tree
1 mark for a tree that is not min.
0 marks for a graph that is not a tree.

(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?

ANSWER
1 mark   DE
1 mark   BC

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.
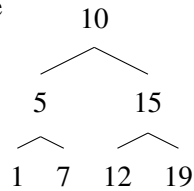
ANSWER

$f(n) = n^3$ is in $\Omega(n)$	T
$f(n) = n^3 \log n$ is in $O(n^3)$	F
$f(n) = n^2 / \log n$ is in $O(n^2)$	T
$f(n) = \log \log n$ is in $O(\log^* n)$	F
$f(n) = \sqrt{n}$ is in $\Omega(\log n)$	T
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	T
An n symbol string that is the result of a Burrows Wheeler Transform can be reversed in $O(n)$ time, regardless of the number of distinct symbols in the string.	F
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 $O(n)$.	T
The travelling salesman problem is in class NP, but the minimum spanning tree problem is in P.	T
Algorithms are still fun	
(no -1 for incorrect on this one; Alistair suggests -3 for incorrect!).	T or F

Turn Over

Question 8 (14 marks).
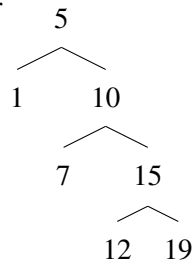
(a) (2 marks) Draw this tree after rotating the root once to the right so that 5 becomes the new root.

ANSWER

```
before        10              after       5
            /    \                       / \
          5       15                    1   10
         / \     /  \                      /  \
        1   7   12   19                   7    15
                                              /  \
                                             12   19
```

1 mark   1 as the left child

1 mark   for 7 as the right-left child

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

ANSWER

Orig	After insert (1 mark)	After rot 1 (2 marks)	After rot 2 (2 marks)

10
 	
5   15
 	
   12 19
	10
	
	5 15
	
	    12 19
	
	  NULL 13
		10
		
		5 12
		
		  NULL 15
		
		     13 19
			12
			
			10 15
			
			5 NULL 13 19

Not necessary to show NULLs, or heights.

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

ANSWER

What is the space required for a hash table that uses Open Addressing and has m slots?	$O(m)$
What is the space required for an AVL tree?	$O(n)$
What is the worst case time to find an item in an AVL tree?	$O(\log n)$
How many bits are required to store a Wavelet Tree based on a prefix code with all codewords of $\lceil \log_2 n \rceil$ bits?	$O(n \log n)$
What is the worst case time required to find the successor of an item in an AVL tree?	$O(\log n)$
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?	$O(m)$

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.

End of the Exam