The University of Melbourne Department of Computing and Information Systems

COMP20007

Design of Algorithms June Assessment, 2014

Student Number:	

Identical Examination papers: None.

Exam Duration: Three hours.

Reading Time: Fifteen minutes.

Open/Closed Book: Closed Book.

Length: This paper has 12 pages including this cover page.

Total Marks: 60

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 (10 marks).

(6 marks) As in Assignment 2, assuming a raster-scan order of grid square visits on a grid, and a Unary code of the actual values, draw the actual grid mapped by the robot the following bitfile?
1 001 00001 001 01 1 01 1 001
(2 marks) In what situation would you use a Unary code to encode symbols? Justify yanswer.
(1 mark) How many bits would be required to code the grid in part (b) using a vbyte of

Question 2 (11 marks).

•	and then define what is meant by a "sink vertex".
	3 marks) What is the minimum number of components in an undirected graph $G(V, E)$ with $ V \ge 2$ if G has one source vertex? Justify your answer.

•		o edges. Note	that this will do	, and returns a count of all uble count each cycle: for
n v through u .	You should as	ssume that G is	stored as an ac	m u through v , and a cycled djacency matrix $G[N][N]$
re N is a parame	eter to your fun	ction, and G[u]] [v] != 0 if th	ere is an edge from u to v

Question 3 (11 marks).

thm that was greedy on weight would not give the optimal answer.
mark) Give a specific example of an input to the Knapsack Problem for which an a

(c) (7 marks) Consider the following recursive algorithm for solving the Knapsack Problem. Find $L(n, n \times \max\{v_i\})$ where L(i, p) is the lowest weight combination of items selected from items 1..i whose total profit is p. If no such combination of items exists, L(i, p) is ∞ . Note item i has integer weight w_i and integer value v_i .

$$L(i,p) = \begin{cases} \infty & \text{if } i = 0\\ L(i-1,p) & \text{if } v_i > p\\ \min\{L(i-1,p), w_i + L(i-1,p-v_i)\} & \text{otherwise} \end{cases}$$

- 1. Give a recurrence relation for the running time of this recursive algorithm.
- 2. Solve the recurrence to get an upper bound on the running time.
- 3. What is the running time of the Dynamic Programming solution based on this recursive algorithm? Why?
- 4. Assuming that the maximum v_i value is $O(\sqrt{n})$, what is the space occupied by an efficient implementation of the Dynamic Programming approach to this algorithm. Justify your answer.

Question 4 (9 marks).

Complete the following table of tight upper bounds on worst case performance for the operation listed in each row for the map data structure in each column. You should assume that the map contains n (key,value) pairs at the time of the operation. You may also assume that hashing and comparing keys requires O(1) time.

	Hash Table with <i>m</i> slots and separate chaining	Binary Search Tree	AVL Tree
Insert			
Find			
Return all values			

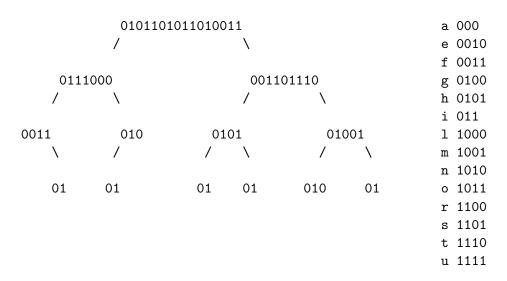
Question 5 (12 marks).

We are interested in comparing implementations of Huffman's algorithm to generate optimal weighted-path trees. An underlying data structure for the algorithm must support two operations: remove-min and insert, where the sum of the leaf weights is the key of each subtree.

TIOW	many internal nodes are there in a Huffman tree with n leaves?
How	many times is $remove-min$ called in Huffman's algorithm for n items?
	many times is $insert$ called in Huffman's algorithm for n items, assuming the original ights are already in the data structure?
const	a tight bound on the worst case running time of Huffman's algorithm when the subtrees tructed during the algorithm are kept in the following data structures. You should justify answer with reference to the previous two answers. You should assume the n input hts are sorted.
1.	A standard binary heap.
2.	A sorted linked list with a finger pointing to the last inserted subtree.
3.	An AVL tree.
	How n we Give const your weight 1.

 COMP20007 Design of Algorithms

Question 6 (7 marks).



(a) What string is represented by this Wavelet Tree where each character in the string is represented by the code on the right? As usual, a 0-bit indicates left and a 1-bit indicates right.

(b) Assuming that rank on an individual bitvector of the tree is answered by simply scanning left-to-right and counting zeroes, how many bits at each level of the tree are inspected to answer the query rank(a, 11) in this tree? (Hint: in the first level of the tree there are 11 bits counted.)

(c) Assuming rank on bitvectors can be done in O(1) time, what is the cost of answering rank on a general alphabet of size σ for a string of length n using a Wavelet Tree?

Overflow Answers

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