THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY
Department of Computer Science

COMP 171 Data Structures and Algorithms
Spring 1996

Final Examination (Section 1)

Derick Wood

Date: May 29, 1996. Time: 08:30–11:30
Place: Sports Hall

You should use the backs of the pages for rough work and, if necessary, for your final answers.

Student name: ________________ **Student ID:** ________________

Problem	Your mark	Maximum	Problem	Your mark	Maximum
1		20	6		15
2		10	7		10
3		10	8		15
4		15	9		15
5		15	10		10
Subtotal		70	Subtotal		65

Your total marks: ____________

1. (20 marks)

Indicate whether the following statements are true or false by circling T, F, or neither. You will obtain two marks for a correct answer, -1 mark for an incorrect answer, and zero marks for no answer.

	True	False	Statement
a.	T	F	Sorting always requires $\Omega(n)$ time in the worst case, where n is the number of records to be sorted.
b.	T	F	Searching using any of the open-addressing hashing methods takes $\Theta(m)$ time in the worst case, where m is the size of the hash table.
c.	T	F	The solution of the recurrence equation $T(0) = T(1) = 1$ and $T(n) = T(n-2) + n$, for $n \geq 1$, is $O(n^2)$.
d.	T	F	Variable-length codes are always prefix codes.
e.	T	F	Given a digraph with n vertices and e edges, the adjacency-list representation takes $O(n+e)$ space.
f.	T	F	Given a digraph with n vertices and e edges, the adjacency-matrix representation requires $\Omega(n+e)$ space.
g.	T	F	Searching, using only key-equality comparisons, takes $\Omega(n)$ time in the worst case, where n is the number of records.
h.	T	F	The solution of the recurrence equation $T(0) = T(1) = 1$ and $T(n) = 2T(n-2) + 1$, for $n \geq 1$, is $O(2^{n/2})$.
i.	T	F	Deletion from binary search tree of size n takes $O(n)$ time in the worst case.
j.	T	F	Multiplying two $n \times n$ matrices takes $O(n^3)$ time in the worst case.

2. (10 marks)
In double hashing each key has two associated values: its hash address and its increment. You are given the following keys

$$24,\ 16,\ 38,\ 76,\ 15,\ 43,\ 30,\ 79,\ 63,\ 49$$

and the following two hash functions that give the hash address and the increment:

$$h(X) = 2X \ \mathbf{mod}\ 11$$

and

$$g(X) = (X\ \mathbf{mod}\ 10) + 1.$$

Insert the keys one at a time into the hash table and show the snapshots after each insertion; we have given you a sequence of hash tables so that you can show the snapshots easily. Give the total number of probes taken by the sequence.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
Probes										

Total probes =________________________

3. (10 marks)
We defined two performance measures IPL and EPL for binary search trees that give the total number of node visits for all keys in a tree and for all gaps in a tree, respectively. In a similar manner, we define Internal Comparison Count (ICC) and External Comparison Count (ECC) as follows:

$$ICC(T) = \sum_{internalnodes\ u} compno(u)$$

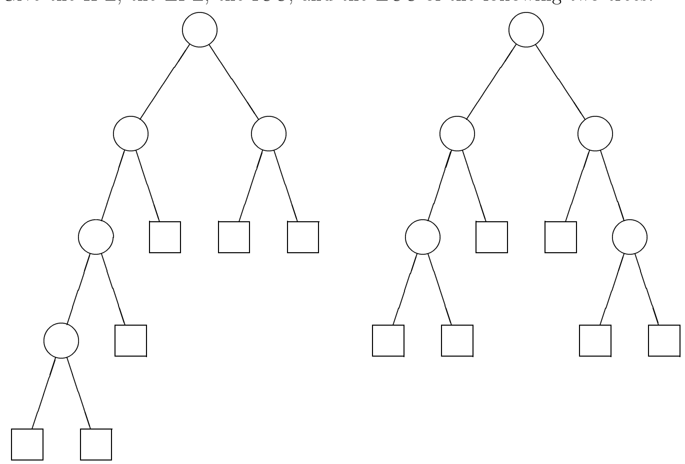
and

$$ECC(T) = \sum_{externalnodes\ u} compno(u)$$

that depend on the number of key comparisons that lead us to each node.

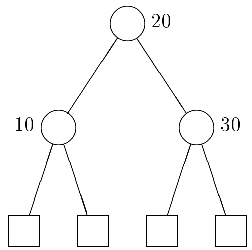
Assume that the search algorithm always first tests whether a search key is less than a node's key and, if it is not, tests for equality: the $(<, =)$ scheme. Then, branching left from a node takes one key comparison, whereas branching right and terminating at a node takes two key comparisons. Detecting that a node is external takes zero key comparisons.

(a) Give the IPL, the EPL, the ICC, and the ECC of the following two trees:



(b) We proved that $EPL(T) = IPL(T) + f(T)$, where $f(T) = 2size(T) + 1$. Now $ECC(T) = ICC(T) + f(T)$, for some $f(T)$. What is the definition of $f(T)$?

4. (15 marks)
You are given the following 2-3-4 search tree T:

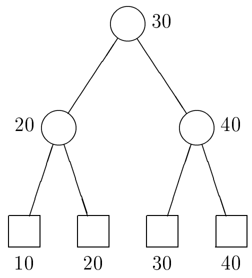


(a) Insert the keys 40, 50, 60, and 70 using the 2-3-4 insertion algorithm. Draw the trees after each insertion is completed.

(b) Delete 30 from T using the 2-3-4 deletion algorithm. Draw the resulting tree after you have completed the deletion.

(c) Delete 30 from T when treating it as a binary search tree. Draw the resulting tree after you have completed the deletion.

5. (15 marks)
You are given the following 2-3$^+$ search tree T:



(a) Insert the keys 50, 60, and 70 using the 2-3$^+$ insertion algorithm. Draw the trees after each insertion is completed.

(b) Delete 10 from T using the 2-3^+ deletion algorithm. Draw the resulting tree after you have completed the deletion.

(c) Delete 10 from T when treating it as an external binary search tree. Draw the resulting tree after you have completed the deletion.

6. (15 marks)
Three questions about digital-search trees.

(a) Given the four binary keys 101, 000, 110, and 001, insert them one at a time into an initially empty binary digital-search tree. Draw the trees after each insertion.

(b) Given the four binary keys 101, 000, 110, and 001, insert them one at a time into an initially empty external binary digital-search tree. Draw the trees after each insertion.

(c) We can use binary search trees to sort keys by first inserting them into an initially empty binary search tree and then doing an inorder traversal of the tree. Can we do something similar with digital-search trees and external digital-search trees? That is, can we sort bit sequences by inserting them into a (external) digital-search tree and doing some kind of traversal? If we can, this would give us another version of radix sort. Please give your answer for each kind of tree and, if your answer is positive, then give a one-sentence statement of what traversal you would use and why.

7. (10 marks) Huffman's prefix-code-construction algorithm works not only with probabilities but also with integer weights. You are given the symbols a, b, c, d, e, f, and g, and the following string:

$c\,c\,b\,a\,a\,c\,d\,a\,e\,d\,d\,d\,b\,d\,b\,e\,a\,e\,a\,c\,c\,d\,a\,a\,d\,b\,e\,a\,c\,e\,a\,c\,c\,d\,a\,a\,a\,a\,d\,e\,d\,e\,b\,e\,b\,a\,c\,a\,a\,d\,e\,d\,e\,b\,e\,b\,a\,c$.

You should assume that the initial frequency of each symbol is one and then add to it the number of times each of the symbols occurs in the string. Give the frequency of each symbol with its singleton code tree. Then, given the forest of these code trees you should compute an optimal binary prefix code using Huffman's algorithm. Show each combination step and the final code tree. Give the expected code length of a symbol as a vulgar fraction, where frequencies divided by (7 + the total number of symbols) replace the probabilities in the original formula for the expected code length.

8. (15 marks) Someone claims that a label sequence they give you is obtained as the preorder traversal of a binary search tree, where the labels are the keys.

   (a) If the sequence is

   $$50\ 20\ 10\ 30\ 40\ 70\ 60,$$

   then is the speaker telling the truth? Justify your answer briefly.

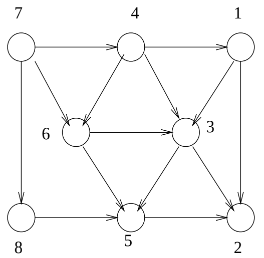
   (b) If the sequence is

   $$50\ 20\ 30\ 10\ 40\ 60\ 70,$$

   then is the speaker telling the truth? Justify your answer briefly.

   (c) Describe an algorithm briefly, in English, to determine whether any given sequence of three distinct integer keys is obtainable with the preorder traversal of a binary search tree for these three keys.

9. (15 marks) You are given the digraph



(a) Do a depth-first traversal of this digraph beginning at vertex 1. To ensure uniqueness of the dft, whenever you have a choice of vertices always choose the vertex with the lowest number. You should draw the depth-first traversal as an orchard making a distinction between tree and examination edges.

(b) Topologically sort the digraph using the following more detailed version of the first inefficient algorithm: *Initialize for each vertex the count of the predecessors it has. Then, for all vertices that have a predecessor count of zero, list them in the order of their vertex numbers. Next, remove all these vertices from the digraph and reduce the predecessor counts of the remaining vertices as necessary. Lastly, repeat this sequence of steps until all vertices are listed.*

To simplify your task give the predecessor counts before each step sequence in the following table and give the topologically sorted order immediately after the table:

	1	2	3	4	5	6	7	8
Initial count								
After Step 1								
After Step 2								
After Step 3								
After Step 4								
After Step 5								
After Step 6								
After Step 7								

T sorted order: ______________________

10. (10 marks) You are the chief software engineer in UST Consulting, a local company. Last week you talked with the software company BEFORE. BEFORE are having efficiency problems with FMS, a file management system, that they had designed, implemented and installed. It is designed to handle large files of 100,000 or more records. It has a well well-designed user interface and it is stable, few errors have been reported. The problem is that although most of BEFORE's customers have no complaints with the system, two of them, A and B, have a serious complaint.

    FMS allows users to initialize a file system with any number of records and do it fast. It provides retrieval on a key and all records have unique keys. It allows insertions of new records and deletion of records that are no longer needed. In addition, it has many other options.

    A and B represent two extreme cases. Company A always has less than 10,000 records and their records are small, less than 100 bytes in size. The search and update time are too slow for them compared to their previous Macrohard system. Company B is the opposite extreme, they are one of a handful of BEFORE customers who are handling more than 100,000 records. They are also complaining about speed.

    After some discussion with BEFORE about FMS, you discovered that it is a disk-based system that uses balanced binary search trees to represent the files.

    What would you suggest that BEFORE should do to alleviate or solve both A and B's problems? State separately for A and B your suggested strategies and justify them. Address the issue of whether you can help both A and B.