Name

CMSC 420 Data Structures

Sample Midterm
Given Spring, 2000
Dr. Michelle Hugue

Problem 1: " Buzzin' in 3-Space" (36 pts)

(6) 1.1 Build a K-D tree from the set of points below using the following assumptions:

- the points are inserted in alphabetical order by label
- the x coordinate is the first key discriminator
- if the key discriminators are equal, go left

A:	(0,7,1)	B:	(1,2,8)	C:	(6, 5,1)
D:	(4,0,6)	E:	(3,1,4)	F:	(4, 0,9)
G:	(9,9,10)	H:	(3,4,7)	I:	(1, 1,1)

(6) 1.2 Explain an algorithm to identify all points within a k-d tree that are within a given distance, r, of the point (x_0, y_0, z_0). Your algorithm should be efficient, in the sense that it should not always need to examine all the nodes.

(6) 1.3 The three-dimensional analog of a point-quadtree is a point-octree. Give an expression for the minimum number of nodes in a *left complete* point-octree with levels numbered $0, 1, \ldots, k$.

(6) 1.4 Explain the difficulties, if any, with deletion of nodes in a point octree. Hint: One way that deletions are performed is by re-inserting any nodes in the subtrees of the deleted one. Why?

(6) 1.5 Describe a set of attributes of the data set, or any other reason, that would cause you to choose a *k-d tree* over a point-octree, and explain your reasoning. Or, if you can't imagine ever choosing the k-d tree instead of the point-octree, explain why.

(6) 1.6 Describe a set of attributes of the data set, or any other reason, that would cause you to choose a *point-octree* over a k-d tree for three dimensional data, and explain your reasoning Or, if you can't imagine ever choosing the point-octree instead of the k-d tree, explain why.

Problem 2:"Ding Dong Bell—Coding Goes Well" (36 pts)

This question addresses issues associated with developing variable length codes specific to known or antici-pated data sets.

A file containing 11 unique symbols is to be encoded using Huffman coding. The frequencies of every letter occuring in the file were tabulated, and the Huffman code in the following table was generated by minimizing the weighted path length in the tree. A fixed length code for the 11 unique symbols was also generated.

Plain Text Letter	Huffman Code Word	4-Bit Code Word
A	100	0001
D	1111101	0010
E	0	0011
I	110	0100
M	11110	0101
O	1111100	0110
R	11101	0111
S	11100	1000
T	101	1001
space	1111110	0000
linefeed	1111111	1111

2.1 (6) Build the Huffman coding tree corresponding to the above code.

2.2 (6) True or False: Letter **A** occurs more frequently in the file than letter **I**. Explain your answer for full credit.

2.3 (6) True or False: The Huffman code for a given set of symbols and frequencies is unique. That is, exactly ONE coding scheme can be extracted from this information. Explain your answer for full credit.

2.4 (6) When decoding a string using a Huffman code, how do you tell when one character stops and another one begins? That is, why don't we need a "new character" or "stop" symbol?

2.5 (6) Is the term "tree" really an accurate name for the Huffman coding tree? That is, is there another name that we could give this structure in the context of the course thus far? Explain your answer for full credit.

2.6 (6) If we receive another file in ASCII containing only those 9 unique symbols and encode it using the table above, which would you expect to produce a shorter encoded file—the 4-bit fixed length code or the Huffman code given above? Explain your answer for full credit.

Problem 3: "Meow Meow: The Revenge of Genghis Kitty" (30 pts)

Last, but not least, here lies a bunch questions that didn't fit elsewhere because Genghis mixed up my pile of notes for the exam. You only have to do **five** (5) of these, not all of them, and if you do more, i'll give you credit for whatever you do.

They are tailored to a variety of skills; so, don't give up *even after* reading them all.

3.1 (6) Genghis claims that the number of leaves in a full binary tree containing n nodes is $(n - 1)/2$ plus 1. Is he correct? Explain for full credit.

3.2 (6) Genghis thinks that search in a binary search tree is a $\theta(n)$ operation. Do you? Explain your answer for full credit.

3.3 (6) Genghis told me that there cannot exist a binary tree yielding the same output order for both preorder and inorder traversals. Is he correct?

3.4 (6) Why did I choose a hybrid data structure, such as the PR quadtree plus B+ tree, for our project? *Genghis thinks it's because I'm a sadist.* Is he right? Or, can you think of a logical reason that I might have done this? That is, what motivates using more than one abstract data type in an application?

3.5 (6) Since Genghis has no front claws, he can't climb a general tree. But, YOU can. Just give me a definition of a general tree, along with on way of representing general trees that supports dynamic allocation of objects of uniform size.

3.6 (6) Genghis is very interested in finding the shortest path out of the house. Let's help him.

Define the *null path length* (npl) of a node in a K-ary tree to be the distance between the node and an empty child. A *leftist tree* satisfies the property that for any node, X, in the binary tree, npl(left_child(X)) is greater than or equal to npl(right_child(X)).

Your job is to explain why a leftist tree rooted at X and having r nodes on the right path must have at least $(2^r - 1)$ nodes.

3.7 (6) Genghis has given up and gone to sleep. However, he left you one last challenge. A dictionary system is maintained and modified heavily throughout the month, and then archived permanently onto a CDROM. Items can be added, deleted, and modified at any time during the month. But, after archiving, only search operations need to be supported.

What structure(s) would you use to implement this dictionary system *throughout* its life cycle? Explain for full credit.