10

Problem 6: "Simple Queries Upset Amazing Seal Hypnotists" (20pt)

Define the Bold Capitalized Terms below in the context of the course. Be brief, but complete.

Term	Definition
6.1) **K-ary Tree:**	A finite set of elements that can have a data element and can have up to K subsets (subtrees) 2
6.2) **Internal Node of a 4-ary tree:**	A node that has at least 1 child node but at max 4 children nodes. nonempty 2
6.3) **Planar Graph:**	A graph that plots line segments that connect two endpoints. ? 2
6.4) **Height of a node in a 3-ary tree :**	1 + the number of edges it takes to reach the deepest root of the tree. 2 ? traditionally single root, multi fine
6.5) **$T(n)$ is in $\Omega(n^{1.5})$:**	$T(n) \geq c \cdot n^{1.5}$ where c is a constant pos n bnt 2

Problem 7: Candidly, Eating Loses Elephant's Red Yarn (40 pts)

We will spend lots of time this semester exploring different techniques for representing spatial data. This problem explores various properties of two of the spatial structures that have made your semester a joy to behold. Be sure to justify or explain your answers for full credit. The following abbreviations will be used below.

- **PO**: point octree
- **KD**: k-d tree of order 3

7.1 Give a brief yet correct definition of a **PO**(point octree).

An 8-ary search tree, that is used to plot 3-d tuples. Dymanic partitioning takes place on a leaf node when two points share the same space.

7.2 Write an expression for the *maximum* number of *keys* or data elements that can be stored in a **PO** (point octree) with the deepest leaf at level q.

$$max = \left(\sum_{i=0}^{q} i \right) + 1$$

7.3 Give an expression for the *minimum* number of *keys* or data elements that can be stored in a **PO** (point octree) with r levels.

$$min = \left(\sum_{i=0}^{r-1} i \right) + 2$$

7.4 What is the asymptotic complexity of searching for a single 3-dimensional key in a **PO**? Briefly explain your answer for full credit.

$O(\log_8 n)$ because it will traverse down the tree based on what appropriate quadrant it may be in and not visit all nodes.

7.5 What is the maximum and minimum number of nodes which must be checked during an unsuccessful `find_key` search in a **PO** containing n keys. Justify your answer for full credit.

4 Max → n because worst case the result is a leaf node of a PO tree with BST disease

Min → 1 because it may be the root

7.6 Give a definition of a k-d tree for three dimensional data, assuming that the grader knows the definition of a k-way search tree for all $k > 1$.

4 A BST that traverses through each comparing keys based on one dimensional attribute. Each level will rotate to the next dim attr. Partitions are made at these attrib

7.7 What is the asymptotic complexity of inserting a new key into a **KD** (k-d tree)? Justify your answer for full credit.

1 O(log n) because its a BST and you compare only logn nodes.

7.8 Which would you expect to be deeper, the **KD** of order 3 or the PO representing the same n keys? Explain your answer.

7.9 Which would you expect to require more space, the **KD** of order 3 or the PO representing the same n keys? Explain your reasoning.

7.10 Give one well explained reason why a **KD** (k-d tree) of order 3 might be preferred over a **PO** for 3-dimensional data. Or, if you don't believe that the **KD** tree would ever be preferred, explain why not.

Problem 8: Just Insulted Crying Alligators–Most Amusing (28pt)

Answer the following questions about **PR** (point-region) quadtrees, being sure to read the questions carefully and justify your work. An example or a picture is a good way to explain your answer and to perhaps earn at least partial credit.
Please assume that all **PR** quadtrees discussed below represent spatial data with integer coordinates and use a $2^x \times 2^x$ grid, where the smallest partition is 1×1. Assume that all quadrants are closed on the left and bottom.

8.1 Give expression for the maximum number of nodes in a **PR** quadtree satisfying the assumptions above.

Gray nodes = $(x-1)$
+
Black nodes = 0

8.2 Give an expression for the maximum number of 2-dimensional coordinate keys which can be inserted in the **PR** quadtree satisfying the assumptions above.

$2^x \times 2^x = \boxed{4^x}$

4

8.3 Briefly explain why the amount of space required by a PR quadtree is in $O(n)$, where n is the maximum number of vertices in the graph it represents.

8.4 Give expressions for `MaxG` the maximum number of gray nodes and `MinG` the minimum number of gray nodes needed by **PR** quadtree which represents a graph having exactly three (3) 2-dimensional coordinate keys. Be sure to label your answers clearly

8.5 Briefly explain why a single `find_key` operation in a PR (point region) quadtree is in $\Omega(1)$.

8.6 Briefly explain why a single `find_key` operation in a PR (point region) quadtree using a 2^x by 2^x grid is in $O(\log_2 n)$ even though the PR quadtree is a 4-way search trie.

8.7 Prove that b, the number of black nodes, in a **PR** is always less than or equal to $3g + 1$, where g represents the number of gray nodes in the **PR**.

10

Problem 9: Tortoise Upsets Rabbit! (Napping Isn't Prudent) (20pt)

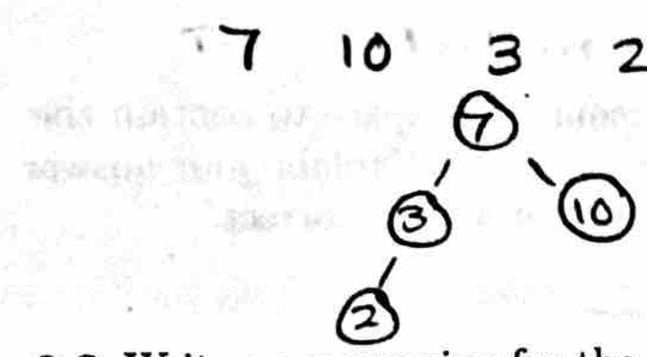
Please answer these questions regarding AVL trees, and explain your work for full credit.

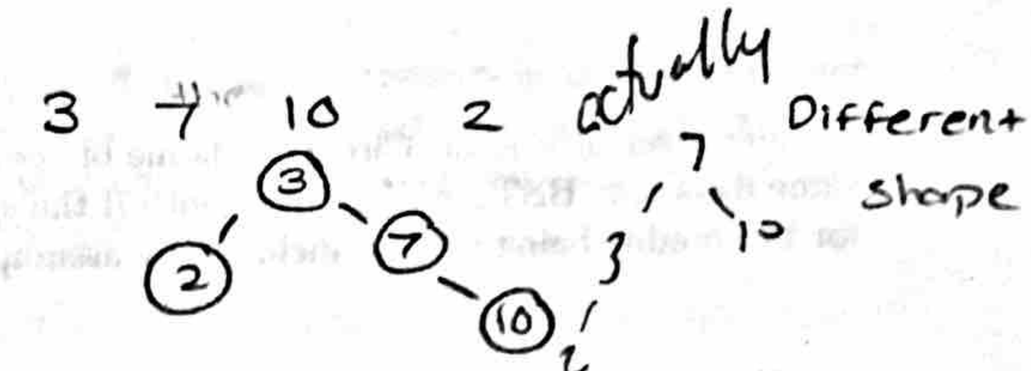
9.1 Give a clear and concise definition of an AVL tree.

2 A BST that has a balancing factor that allows the left and right tree to differ by 9 (usually 1). If it violates, it rebalances itself using single / double rotations

9.2 Prove or Disprove: The shape of the AVL tree created by inserting n keys is the same, regardless of the order in which the keys are inserted.

2

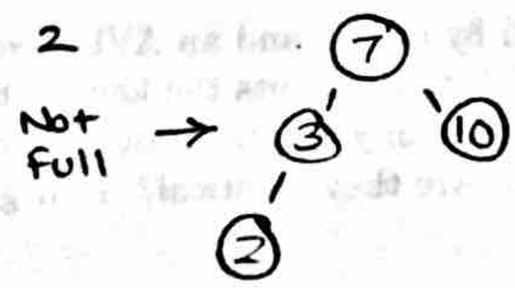




9.3 Write an expression for the maximum number of keys in an AVL tree having $p > 0$ levels. Be sure to show your work!

9.4 Prove or Disprove: All AVL trees are full binary search trees..

4



But still considered AVL

9.5 Prove or Disprove: The number of leaves in an AVL tree with n keys is $L = 2n + 1$.

9.6 Write an expression for the maximum number of keys in an AVL tree where the deepest leaf is at level y. Your final answer will contain no summation symbols ($\sum$). Feel free to circle your final answer to assist the grader.

9.7 Suppose that you have constructed an AVL tree and a BST from a given set of n keys. Which would you expect to be higher–the AVL tree or the BST? Justify your answer for full credit.

2 BST is higher because AVL balances itself to a height of $\log_2(n)$ and the data could potentially have a lot of nodes that have one empty child

9.8 Consider a set of n keys. Circle the name of the structure which will require less space to contain the same data set: BST, AVL. Circle both if the space requirements are the same. Explain your answer for full credit, being sure to include any assumptions required to make your answer correct.

9.9 Once again, consider a set of n keys represented by a BST and an AVL tree. Suppose that an in-order traversal (Left, Process Root, Right) of the AVL tree outputs the keys in the same order as an in-order traversal of the BST. What, if anything, can you infer about the relationship between the BST and AVL tree? Are they the same shape? Are they identical? Your grade will be based on the justification of your answer here.

Problem 10: "Osculating Koi Romancing Aquatically " (20pts)

There are some practical application-ish question here. Read the questions carefully, and don't be afraid to use common sense.

10.1–10.2 Suppose that the *user manual* for a new piece of software uses the following definitions in its section on *database query documentation for users.*

```
Insert_Key:  create a new data record associated with the new key after an
unsuccessful find_key query.  Modify the primary key index and the data
record database appropriately.

Delete_Key:  remove the data record from the database after a successful
find_key query.
```

What attributes associated with the program or how it manages its resources, or how this *user manual* is written, can you *reasonably* infer from these two definitions? I thought of four; so, I'll award full credit of 4 points for each of two different ones, as long as you explain how the *supplied information* provides sufficient context to support your inferences.

10.3 You are given a data file that you know contains an n-node binary tree. The keys (data objects) in the nodes are comparable; furthermore, you know that a unique sorted ordering is associated with the set of keys in the data file taken in ascending order.

Describe an algorithm in clear, concise, English that takes a binary tree as input and returns true if the binary tree is also a valid **BST**.

For full credit, do not write pseudocode. **do not** write a JAVA or C++ or C or python procedure, or any other programming language procedure.

10.4 Let $T(n)$ and $W(n)$ represent the execution times of two different algorithms. If $T(n) \in O(n)$ and $W(n) \in O(n)$, what, if anything, can you conclude about the relationship between $T(n)$ and $W(n)$? Justify your answer for full credit.

10.5 If the execution time, $T(n)$, of an algorithm is known to be in $O(\log_4 n)$, can we always find positive constants μ and ψ such that $T(n) = \mu \log_4 n + \psi$? Justify your answer for full credit.

Extra Credit What do the words below have in common? Illustrate your answer by including another word that belongs in the set.

to	fro	stall	mar
are	tort	leo	ion
card	rob	goo	rave
pan	buff	hip	her
man	ale	sea	ark