1: General Terminology

ints) The following are questions about terms used in our course.

Our course name is abbreviated to ADAA. What does ADAA stand for?

We have looked at ADTs and data structures to analyze run-time. What does ADT stand for?

Abstract data type.

Which Java IDE are we using for programming and homework submission?

What is your instructor's last name?

tion 2: Data

(4 points) The following are True/False questions about data in Computer Science. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓣ Ⓕ An ADT defines the behavior of a data implementation from the point of the ***user***.

Ⓣ Ⓕ A data structure is a representation of the data (organization, storage, and management) from the point of the ***programmer***.

Ⓣ Ⓕ An ADT can have several data structures associated with it.

Ⓣ Ⓕ An ADT is implementation-dependent.

(4 points) The following are questions about particular ADTs / data structures. Mark either ADT or DS to indicate which one is applicable.

ADT DS Hash Table

ADT DS List

ADT DS Binary Search Tree

ADT DS Stack

4. (4

aps and Trees

binary heap is a nearly complete binary tree filled on all levels except possibly the lowes
e leaves are pushed left-most. Heaps are often implemented as an array.
n array A that implements a heap is usually of length power-of-two. If a heap has size 126
ements, what is the (minimum) size of the corresponding array?

$2^{\log_2(126)}$

b. Each node in a heap satisfies the heap property. What is the name of a heap in which every node's value is smaller than the values of its children?

min heap

c. If a binary heap contains 28 elements (nodes), what is the height of the corresponding binary tree?

$\log_2 28$

d. If a heap is implemented in an array A, and the maximum in that heap is the first element of the array, what type of heap is it?

Max heap

5. (4 points) A 2-3 tree is a tree in which each non-root node which is not a leaf has 2 or 3 sons. The following are True/False questions about 2-3 trees. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓣ Ⓕ Each node is labeled with the smallest value in the left subtree and the largest value in the middle subtree.

Ⓣ Ⓕ Every path from the root to a leaf has length 2 or length 3.

Ⓣ Ⓕ Data is stored only in leaves.

Ⓣ Ⓕ Data is ordered left-to-right.

5. (4 points) A binary search tree (BST) is a linked-node based binary tree which stores key-value pairs (or just keys) in each node. Left and right children are roots of left and right subtrees, respectively. The following are True/False questions about BSTs. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓣ Ⓕ BSTs form doubly linked lists.

Ⓣ Ⓕ All keys of nodes in the right subtree of a node N are smaller than the key of N.

Ⓣ Ⓕ Keys in a BST must be integers.

Ⓣ Ⓕ The minimum key in a BST is in the root.

max heap

7. (4 points) A binary search tree (BST) is a linked-node based binary tree which stores key-value pairs (or just keys) in each node. Left and right children are roots of left and right subtrees, respectively. The following are True/False questions about BSTs. Mark either Ⓣ (for True) or Ⓕ (for False).

   Ⓣ Ⓕ The size of the left subtree must be larger than the size of the right subtree since nodes must be pushed left.

   Ⓣ Ⓕ Post-order walks provide the correct key order regardless of the tree balance.

   Ⓣ Ⓕ A BST with n nodes has height at least $\approx \log_2(n)$.

   Ⓣ Ⓕ Keys in a BST must be comparable.

8. (4 points) A binary search tree (BST) is a linked-node based binary tree which stores key-value pairs (or just keys) in each node. Left and right children are roots of left and right subtrees, respectively.

   a. What is the relationship between the key of a parent and the key of its left child?

   node.key>= node.key.child

   b. Give a short description of an algorithm to find the successor of a node N (by key) after the node N has been located. No programming on paper!

Section 4: Self-Balancing Trees and Forests

9. (4 points) A self-balancing tree (forest) is a (collection of) search tree data structure(s) in which insert/delete operations may trigger a partial tree rebuild. **Name four** self-balancing search tree data structures we discussed in class or have been assigned as presentation topics.

   a.

   b.

   c.

   d.

10. (4 points) Scapegoat trees are search trees which upon insert/delete operations rarely but expensively choose a scapegoat node and completely rebuild the subtree rooted at it into a complete tree. The following are True/False questions about Scapegoat trees. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓕ Scapegoat trees are binary search trees.

Ⓕ Scapegoat trees store the height of the whole tree in the root node.

Ⓕ Scapegoat trees store the height of the subtree rooted at a node N in that node N.

Ⓕ Scapegoat trees store the weight of the subtree rooted at a node N in that node N.

11. (4 points) Scapegoat trees are search trees which upon insert/delete operations rarely but expensively choose a scapegoat node and completely rebuild the subtree rooted at it into a complete tree. The following are True/False questions about Scapegoat trees. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓕ A measure of tree balance is the parameter α. For a Scapegoat tree, size(left[node]) ≤ α·size(node).

Ⓕ A measure of tree balance is the parameter α. For a Scapegoat tree, $\frac{1}{2} \leq \alpha \leq 1$.

Ⓕ If a partial tree rebuild is triggered by insertion of a deep node N, the scapegoat node is an ancestor of the node N.

Ⓕ If T is an α-weight-balanced binary search tree then T is also α-height-balanced.

12. (4 points) A priority queue is a special type of queue in which each element is associated with a priority value. Elements are served on the basis of their priority. Higher priority elements are served first. Elements with the same priority are served according to their order in the queue. Priorities can be encoded with keys.

Name two algorithms or applications for which priority queues are used.

a.

b.

Name two data structures that we looked at for implementation of a priority queue in class.

c.

d.

13. (4 points) Fibonacci heaps are a collection of trees. The following are True/False questions about Fibonacci heaps. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓕ The roots of the trees in a Fibonacci heap are stored in a doubly linked list.

Ⓣ Fibonacci heaps consolidate trees after each DELETE_MIN operation.

Ⓕ Node in a Fibonacci heap have parent pointers.

Ⓣ Ⓕ A node N in a Fibonacci heap has pointers to each of its children.