ection 1: General Terminology

(4 points) The following are questions about terms used in our course.

a. We covered several data structures and associated algorithms in this course. Which structures (not forests) did we cover?

b. The co-creator of Scapegoat trees is one of the authors of our text book. He is famous in his own right for his contribution to cryptography. What is his name?

Ronald L Rivest

c. Thirteen groups of students submitted group presentations. What was your group's topic?

d. Nodes which are referenced in a linked data structure may be missing. In that case, they must be referenced by a particular object. What is the word used in Java to indicate a missing link?

ction 2: Data

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

Ⓕ An ADT is implementation-independent.

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

Ⓕ An ADT defines the behavior of a data implementation and serves as an API to the *user* of the API.

Ⓕ A data structure is a representation of the data (organization, storage, algorithms, and management) and is usually hidden from the *user* of the API.

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

DT DS String ADT

DT DS java.util.ArrayList

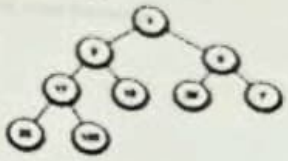
T DS Deque ADT

T DS java.math.BigInteger ADT

Section 3: Heaps and Trees

4. (4 points) A binary heap is a nearly complete binary tree filled on all levels except possibly the lowest level where leaves are pushed left-most. Heaps are often implemented as an array.

   a. The given heap is represented as a binary tree with integer key entries. What would its representation in array-form look like? Write an array with nine entries in the correct order?



   b. If a binary heap contains 60 elements (nodes), what is the height of the corresponding binary tree?

$h = \log_2(60)$

   c. If a binary heap contains 126 elements (nodes) and is represented as an array A, what is the index in A of the root element in the corresponding binary tree?

zero

   d. What are the minimum and maximum numbers of elements in a heap of height h?

$2^h, 2^{h+1} - 1$

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 largest value in the left subtree and the largest value in the middle subtree.

   Ⓣ Ⓕ Data is stored in any non-root nodes.

   Ⓣ Ⓕ Data is ordered left-to-right.

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

6. (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 can take on the form of a linked list where each node has only one child.

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

   Ⓣ Ⓕ The minimum in a BST can be found by following the left child pointers from root until we encounter a leaf.

   Ⓣ Ⓕ The size of a left subtree and a right subtree differ by at most one to balance the tree.

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).

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

Ⓣ Ⓕ The minimum key is in the root.

Ⓣ Ⓕ A BST with n nodes has height at most n.

Ⓣ Ⓕ Keys in a BST must be integers.

Section 4: Self-Balancing Trees and Forests

8. (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 subtree rooted at a node N in that node N.

Ⓣ Ⓕ Scapegoat trees are lazily height balanced trees.

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

9. (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, $\frac{1}{2} \leq \alpha \leq 1$.

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

Ⓣ Ⓕ A measure of tree balance is the parameter α. For a Scapegoat tree, $\alpha \cdot$size(node) $\leq$ size(left[node]).

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

10. (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).

Ⓣ Ⓕ Fibonacci heaps consolidate trees after each INSERT operation.

Ⓣ Ⓕ Fibonacci heaps have better asymptotic bounds than binary heaps for INSERT, UNION, and DECREASE_KEY, and have the same asymptotic running times for the other operations.

Ⓣ Ⓕ Roots of trees in a Fibonacci heap are stored in a doubly-linked list.

Ⓣ Ⓕ Each child in a tree of a Fibonacci heap has a parent pointer.

Section 5: Runtime Analysis, Amortized Analysis, and Lower Bounds

11. (4 points) Hash tables are an implementation of the ADT dictionary. They are often used to store functions.

a. What do we call it when two keys are mapped to the same hash value?

Collision

b. When two keys are mapped to the same hash value, what is the name of the method to store data in the corresponding array entry?

Chaining

c. If we implement a hash table as an array of size 256, what does the output of our hash function $Hash(x)$ have to satisfy (related to the definition of a hash function) in order for our program not to constantly crash?

Fixed o/p size, no collision

d. Keys for arrays must be comparable. If two keys $k_1 < k_2$ are hashed, what can you say about the comparison of their hash values $Hash(k_1)$ and $Hash(k_2)$?

? Hash(k_1) ≠ Hash(k_2)

12. (4 points) Run-time analysis is an estimation of running time of an algorithm as a function of its input size (usually denoted as n). The following are four True/False questions about runtime analysis. Mark either Ⓣ or Ⓕ.

Ⓕ In a BST with n nodes, the BST key property affords us to retrieve all data in order with a recursive walk in $O(n)$.

Ⓣ Ⓕ FIND/SEARCH/GET in a BST with n nodes and height h always has runtime of $O(\log(n))$.

Ⓣ Ⓕ FIND/SEARCH/GET in a 2-3 tree with n nodes always has runtime of $O(\log(n))$.

Ⓣ Ⓕ FIND/SEARCH/GET in an array with n keys always has runtime of $O(1)$.

13. (4 points) Amortized analysis is a method for analyzing an algorithm's complexity. The following are four True/False questions about amortization analysis. Mark either Ⓣ or Ⓕ.

Ⓣ Ⓕ Amortization is used for the evaluation of a sequence of operation.

Ⓕ Amortized analysis usually gives better upper bounds on the running time than traditional analysis.

Ⓣ Ⓕ Amortized analysis evaluates the average cost.

Ⓕ Scapegoat trees achieve $O(\log(n))$ amortized run-rime complexity for all operations INSERT, DELETE, SEARCH.

14. (4 points) The following are True/False questions about information theoretic lower bounds. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓣ Ⓕ The length of the longest simple path from the root of a decision tree to any of its reachable leaves represents the worst-case number of comparisons that the corresponding sorting algorithm performs.

Ⓣ Ⓕ The complexity of an algorithm A which solves a problem P is a lower bound on the complexity of P.

Ⓣ Ⓕ Ω provides an asymptotic lower bound.

Ⓣ Ⓕ Θ provides an asymptotically tight bound.

Section 6: Graph Theory and Flow Networks

15. (4 points) A graph traversal is a systematic procedure for exploring a graph by examining all of its vertices and edges. Name two standard graph traversal algorithms.

a. B.F.S

b. D.F.S

Name two standard data structures to represent graphs (the vertex-edge relationships) in computer science.

c. Adjacency Matrix

d. || List

16. (4 points)

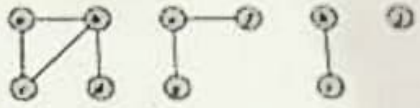
a. What is the name of the Python library we used for graphs in class and on homework?

igraph

b. In graph theory, what is the name of a map $f(V) \to V$ from the vertices of a graph to itself such that edges get mapped to edges (incidence is preserved)?

Automorphism

c. The given graph has ten vertices and four connected components. Which algorithm did we use to find connected graph components (sets of vertices) in class and on homework?



d. The final homework dealt with graph symmetries of a provided graph. How many digits did the number of symmetries of this graph have (or, if you remember, what was the number)?

17. (4 points)

a. In a flow network, what is the name of a node that has only incoming flow?

sink

b. In a flow network, what is the name of a node that has only outgoing flow?

source

c. In a flow network, what is the graph theoretical term for "what comes in must go out"?

flow conservation

d. Which relation to zero is correct for a network capacity $c(u, v)$ from vertex u to vertex v in a flow network? **Mark only one!**

- ○ $c(u, v) < 0$
- ○ $c(u, v) \leq 0$
- ○ $c(u, v) = 0$
- ○ $c(u, v) \neq 0$
- ○ $c(u, v) \geq 0$
- ○ $c(u, v) > 0$

18. (4 points) The following are True/False questions about flow networks. Mark either Ⓣ (for True) or Ⓕ (for False).

Ⓣ Ⓕ A flow with the largest possible value must exist.

Ⓣ Ⓕ No flow may exceed the capacity along an edge or path.

Ⓣ Ⓕ A greedy algorithm chooses the path with the initially largest capacity from the starting point to the end. Greedy algorithms provide an optimal solution for the maximum flow problem.

Ⓣ Ⓕ Network capacity is the minimum amount of flow that can be reliably transferred between different locations over a flow network.