Section 1: General Terminology

1. (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 was the first tree structure we looked at?

   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. In your homework on Scapegoat trees you overrode a method of the Scapegoat class so you could print out the data in order. What is the name of the Java method to override so you change the way `System.out.println` prints an object?

   toString method

Section 2: Data

2. (4 points) The following are True/False questions about data in Computer Science. Mark either Ⓣ (for True) or Ⓕ (for False).

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

   Ⓣ [✓ Ⓕ] A data structure is a representation of the data (organization, storage, algorithms, and management) and needs to accessible to the **user** of the application programming interface (API).

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

   [✓ Ⓣ] Ⓕ An ADT is implementation-independent.

3. (4 points) The following are questions about particular ADTs / data structures. Mark either ADT or to indicate which one is applicable.

   ADT [✓ DS] Binary Search Tree

   ADT [✓ DS] java.util.ArrayList

   [✓ ADT] DS Stack

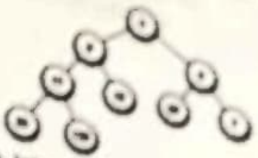
   [✓ ADT] DS String

ction 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. Each node in a heap satisfies the heap property. What type of heap is the given binary heap (represented as a binary tree with integer key entries)?

   Min heap



   b. The given heap in (a) is represented as a binary tree with integer key entries. When represented in array form, these nine entries would be written in consecutive order. Which key would be the last array entry?

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

   $h = \log_2(60)$

   d. What are the minimum and maximum numbers of elements in a heap of height 10?

   $2^h, 2^{h+1} - 1 \Rightarrow 2^{10}, 2^{11} - 1$

(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 middle subtree and the largest value in the right subtree.

[✓ Ⓕ] Data is stored in any non-root nodes.

Ⓕ Data is ordered left-to-right. T

Ⓕ Every path from the root to a leaf is of the same length. T

oints) A binary search tree (BST) is a linked-node based binary tree which stores key-value
st keys) in each node. Left and right children are roots of left and right subtrees, respecti
ollowing are True/False questions about BSTs. Mark either Ⓣ (for True) or Ⓕ (for False).

[✓ Ⓕ] Every node in a BST has two children.

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

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

The size of a left subtree and a right subtree differ by at most one to balance t

F

) A binary search tree (BST) is a linked-node based binary tree which stores key-value
(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).

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

Ⓣ [✓ Ⓕ] The maximum key is in the root.

[✓ Ⓣ] Ⓕ A BST with n nodes has height $= \log_2(n)$.

[✓ Ⓣ] Ⓕ Keys in a BST must be comparable.

Section 4: Self-Balancing Trees and Forests

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

Ⓣ Ⓕ Scapegoat trees are specialized 2-3 trees.

[✓ Ⓣ] Ⓕ Scapegoat trees are lazily height balanced trees.

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

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

(4 points) Scapegoat trees are search trees which upon insert/delete operations rarely but expensive
choose a scapegoat node and completely rebuild the subtree rooted at it into a complete tree. Th
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, $\alpha = 0.55$.

[✓ Ⓣ] Ⓕ If T is an α-weight-balanced binary search tree then T is also α-height-balanced

Ⓣ Ⓕ Scapegoat trees calculate the height of an inserted node N on the fly to determine
whether it is a deep node.

Ⓣ [✓ Ⓕ] If a partial tree rebuild is triggered by insertion of a deep node N, the scapegoat node is a
descendant of the node N.

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

[✓ Ⓣ] Ⓕ Fibonacci heaps were created to minimize the number of operations needed to compute
Minimum Spanning Trees.

Ⓣ [✓ Ⓕ] Roots of trees in a Fibonacci heap are stored in a singly-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 arrays which utilize hash functions.

    a. 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

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

    Collision

    c. If we implement a hash table as an array of size 1024, 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 output size, no collision

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

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 $O(\log^2(n))$.

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

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

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 Ⓕ.

    [✓ Ⓣ] Ⓕ Amortized analysis evaluates the average cost.

    [✓ Ⓣ] Ⓕ Scapegoat trees achieve $O(\log(n))$ amortized run-time complexity for all operations INSERT, DELETE, SEARCH.

    [✓ Ⓣ] Ⓕ 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.

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.

    Ⓣ [✓ Ⓕ] O provides an asymptotic lower bound.

    Ⓣ [✓ Ⓕ] Ω provides an asymptotically tight bound.

Section 6: Graph Theory and Flow Networks

15. (4 points) Name two standard data structures to represent graphs (the vertex-edge relationships computer science.

a. Adjacency Matrix

b. Adjacency List

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.

c. Breadth First Search

d. Depth " "

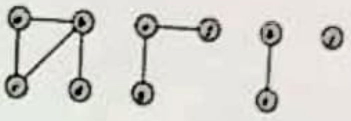
16. (4 points)

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

igraph

b. 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)?

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

automorphism

7. (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!

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