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 data structure/lecture did we cover the week after Thanksgiving?

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

   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 in Python to indicate a missing link?

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 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 always visible to the* ***user*** *of the API.*

   Ⓣ Ⓕ *An ADT has exactly one data structure associated with it.*

   Ⓣ Ⓕ *An ADT is implementation-independent.*

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

T DS *Tree*

DS *Scapegoat Tree*

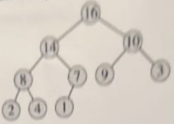
DS *java.util.ArrayList*

S *Stack*

...and Trees

...heap is a nearly complete binary tree filled on all levels except possibly the lowest ...are pushed left-most. Heaps are often implemented as an array.

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



...ary heap contains 126 elements (nodes) and is represented as an array A, what is the ... of the corresponding binary tree?

...heap is implemented as an array, at which index is the minimum in a min-heap located?

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

...ts) A 2-3 tree is a tree in which each non-root node which is not a leaf has 2 or 3 sons. The ...ing 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 right subtree.
- Ⓕ Data is stored in all non-root nodes.
- Ⓕ A 2-3 tree has the min-heap property.
- Ⓕ Every path from the root to a leaf has length 2 or 3.

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

- Ⓕ A BST with n nodes has at most height $\approx \log_2(n)$.
- Ⓕ 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 the left subtree and the size of the right subtree must be the same.

(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 maximum key is in the root.

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

Ⓣ Ⓕ Keys in a BST must be integers.

ction 4: Self-Balancing Trees and Forests

(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 store the weight of the subtree rooted at a node N in that node N.

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

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

Ⓣ Ⓕ Scapegoat trees are binary search trees.

(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).

Ⓣ Ⓕ 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, α·size(node) ≥ size(left[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 a descendant of the node N.

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

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

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

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.

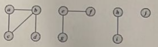
    Name two standard data structures to represent graphs (the vertex-edge relationships) in computer science.
    c.

    d.

16. (4 points)
    a. 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)?

    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)?

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

17. (4 points)
    a. In a flow network, what is the name of a node that has only outgoing flow?

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

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

    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!**

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

3. (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 may not 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 sometimes do not 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.

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 algorithm/lecture is Dr. Fiedler's favorite (which he worked with during his post-doctoral time and considers the most important algorithm for today's lifestyle)?

   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?

   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?

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 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.

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

   Ⓣ Ⓕ An ADT is implementation-dependent.

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

   ADT DS Binary Search Tree

   ADT DS List

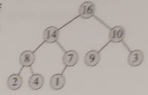
   ADT DS Integer

   ADT DS Dictionary

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. 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)?

   b. The algorithms (insert, delete, find) for a binary heap represented by an array A of integer keys need additional attributes to function properly (so they do not return wrong results or access violations). What are these attributes?

   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?

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

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 only in leaves.

   Ⓣ Ⓕ Data is ordered left-to-right.

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

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 right subtree of a node N are smaller than the key of N.

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

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

11. (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 are named due to their tree sizes being bounded by Fibonacci numbers.

Ⓣ Ⓕ All children in a tree of a Fibonacci heap have links to their parents.

Ⓣ Ⓕ Fibonacci heaps consolidate trees after each INSERT operation.

Ⓣ Ⓕ Fibonacci heaps optimize INSERT operations.

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

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 Ⓕ.

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

Ⓣ Ⓕ 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 $O(1)$.

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

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 Ⓕ.

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

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

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

Ⓣ Ⓕ Amortized analysis evaluates the average cost.

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 best-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.

Ⓣ Ⓕ Θ provides an asymptotic lower bound.

Ⓣ Ⓕ O provides an asymptotic upper bound.

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 at most height $\approx \log_2(n)$.

Ⓣ Ⓕ 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 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.

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

Ⓣ Ⓕ Scapegoat trees are binary search trees.

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).

Ⓣ Ⓕ 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]).

Ⓣ Ⓕ A measure of tree balance is the parameter α. For a Scapegoat tree, $0 \leq \alpha \leq \frac{1}{2}$.

Ⓣ Ⓕ 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) 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.

8. (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 maximum amount of flow that can be reliably transferred between different locations over a flow network.

Section 7: Graph Theory and Flow Networks

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

    a.

    b.

    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.

    d.

16. (4 points)

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

    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)?

    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?

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

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

    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!**

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

11. (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 are named due to their tree sizes being bounded by Fibonacci numbers.

Ⓣ Ⓕ Fibonacci heaps consolidate trees after each INSERT operation.

Ⓣ Ⓕ Fibonacci heaps optimize DECREASE_KEY operation.

Ⓣ Ⓕ Siblings in a tree of a Fibonacci heap are doubly-linked.

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

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 $O(\log(n))$.

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

Ⓣ Ⓕ 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.

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

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

Ⓣ Ⓕ Amortized analysis evaluates the average cost.

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 an **upper** bound on the complexity of P.

Ⓣ Ⓕ Θ provides an asymptotic lower bound.

Ⓣ Ⓕ Ω provides an asymptotically tight bound.