0

Illinois Institute of Technology Computer Science

Final Exam

CS 331 — Data Structures
Fall, 2010
2010-12-09 08:00–10:00

This is a **closed book** and **closed notes** exam.
You are **not** allowed to use calculators or computers during this exam.
Do **ALL** problems in this booklet. Read each question very carefully.
You may detach pages, but **you must return all pages of this exam.**

Name	
Email ID	@iit.edu

Problem	Points	Score
Total	0	
Percent	100	

1 Abstraction

Question 1) (4 points) What is an abstract data type?

Question 2) (4 points)
What are two advantages of using abstract data types?

Question 3) (4 points)
You are the manager for two programming teams. You ask them to design a stack class for your software product's interface.

The first team noticed that the list class already included some extra methods like push and pop, so they simply copied the list class, naming it `stack`, and used it everywhere that they needed a stack. This approach saved a couple of weeks of programming time. The second team created a brand new stack class, and used an instance of the list as a private variable. They provided accessor methods for the methods they needed, but left out the rest of them. This drastically reduced the functionality of the stack class by leaving out most of the list methods.

Which team did the right thing, and why?

2 Memory

Question 4) (4 points) Consider the code below.

```
public class Foo {
  Foo x;
  int y;
  Integer z;
}

Foo a = new Foo();
Foo b = new Foo();

a.x = a;
a.y = 20;
a.z = new Integer(30);

b.x = a;
b.y = 20;
b.z = new Integer(30);
```

Draw a memory diagram that illustrates the effect of the above code.

Question 5) (4 points) Now show the effect of the code

```
b.x = b;
b.z = a.z;
```

Indicate which memory, if any, becomes garbage.

3 Lists

Question 6) (4 points)

Under what circumstances would you prefer an array implementation of a list to a linked implementation?

Question 7) (4 points) Some implementations of singly linked lists have a `last` pointer to point to the last element of a list. This is done to speed up operations that involve the end of the list, such as `insertAtEnd`. It does not help `removeFromEnd` though. Explain why it doesn't help.

4 Locality

Question 8) (4 points) What is temporal locality?

Question 9) (4 points) Under what circumstances would you use the *swap* heuristic instead of the *move-to-front* heuristic?

Question 10) (4 points)
Consider this list.

Suppose we now find the 8 node and then the 3 node. Show the effect on the list if we use *move-to-front* heuristic.

5 Stacks

Question 11) (4 points) What are the names of the three stack operations? Give their time complexities.

Question 12) (4 points) Suppose you have a singly-linked `List` class, with a last pointer, like the one we've discussed in class. Suppose also that we have used this as our implementation for a stack. Write the code for the `pop` method.

6 Queues

Question 13) (4 points) What are the names of the three queue operations? Give their time complexities.

Question 14) (4 points) Suppose you have a singly-linked `List` class, with a last pointer, like the one we've discussed in class. Suppose also that we have used this as our implementation for a queue. Write the code for the `dequeue` method.

Question 15) (4 points)

In class we discussed how to implement a queue efficiently using linked lists when the list implementation does not have a last pointer. Briefly describe this implementation. (No coding, just English.)

7 Doubly Linked Lists and Friends

Question 16) (4 points)

Write a function `deleteNode(Node n)` that deletes node `n` from a doubly-linked list. Indicate whether or not you are using sentinels. Element `n` is guaranteed to be in the list. Your code must run in $\mathcal{O}(\infty)$ time to get credit.

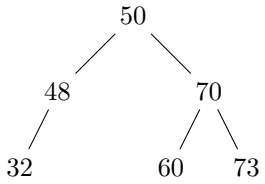
Question 17) (4 points)

Describe a method for determining the tower height of a node in a Skip List.

8 Binary Search Trees

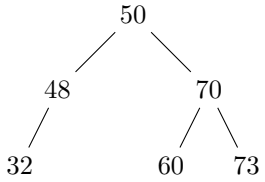
Question 18) (4 points)

Show the effect of adding a 38 and a 34 to the following tree. You may modify the figure or redraw it.



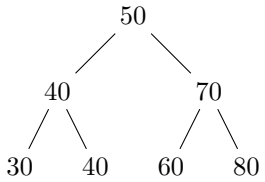
Question 19) (4 points)

Show the effect of deleting element 48 from the following tree. You may modify the figure or redraw it.



Question 20) (4 points)

Show the effect of deleting element 50 from following tree. You may modify the figure or redraw it.

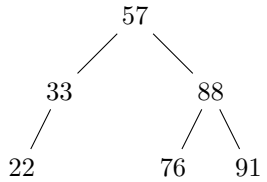


Question 21) (4 points)
What is the expected time complexity for a find operation on a BST?

Question 22) (4 points) What is the worst case time complexity for find? Under what circumstances does it occur?

9 Traversals

Question 23) (8 points)



Write down the following traversals for the above tree:

Preorder

Postorder

Inorder

Level-order

Question 24) (4 points) Assuming you have a balanced tree, how much memory will a breadth-first traversal require?

10 Sorting

Question 25) (4 points)

Two sorting algorithms we studied are able to provide useful data even if they are interrupted. Which are they?

Question 26) (4 points)

What does it mean if a sort is said to be *stable*?

Question 27) (6 points)

One sorting algorithm suffers an extreme loss of performance if the data it is given happen to be already sorted. Which algorithm is this, and how can the algorithm be fixed to prevent this?

Consider the following array:

126	122	219	121	317	123	114

Question 28) (4 points)
What will be the first three elements selected if we run selection sort on this array?

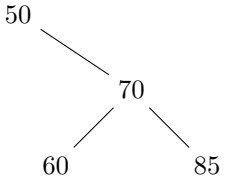
Question 29) (4 points)
What will be the first three elements inserted if we run insertion sort on this array?

Question 30) (4 points)
Show the effect of quicksort's partition algorithm on the above array. Use the first element as the pivot.

11 AVL Trees

Question 31) (6 points)

Consider the following AVL tree.



A node has just been deleted from the tree, and now it is out of balance.

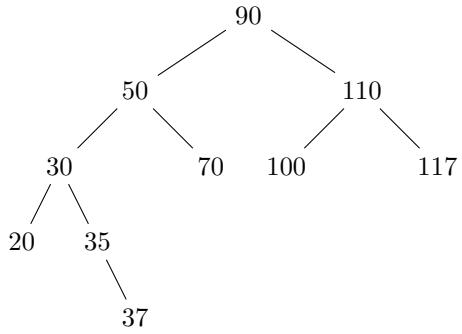
1. Which node is out of balance?

2. What kind of rotation is needed to restore balance?

3. Show the effect of the rotation.

Question 32) (6 points)

Consider the following AVL tree.



A node has just been added to the tree, and now it is out of balance.

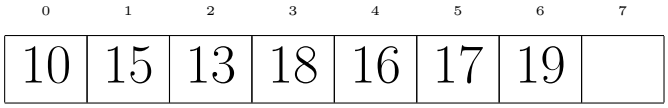
1. Which node is out of balance?

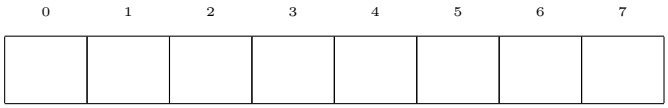
2. What kind of rotation is needed to restore balance?

3. Show the effect of the rotation.

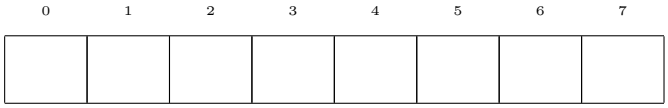
12 Heaps

Question 33) (4 points) Given the following heap, show the result of deleting two elements from it.





(After one delete)



(After two deletes)

Question 34) (4 points) What is the time complexity of deleting an element from a heap?

13 Hash Tables

Consider the following table:

x	$h_1(x)$	$h_2(x)$
apple	3	5
banana	3	3
cherry	3	4
durian	3	2

Question 35) (4 points)
Show the effect of hashing the data (in alpha order) into this hash-table using separate chaining.

i	$t[i]$
0	
1	
2	
3	
4	
5	
6	

Question 36) (4 points)
Show the effect of hashing the data (in alpha order) into this hash-table using linear probing.

i	$t[i]$
0	
1	
2	
3	
4	
5	
6	

Question 37) (4 points)
Show the effect of hashing the data (in alpha order) into this hash-table using double hashing.

i	$t[i]$
0	
1	
2	
3	
4	
5	
6	

Question 38) (4 points) What is the expected time complexity of a hash-table insertion, given a reasonably empty hash table?

Question 39) (4 points) What is the expected time complexity of a hash-table insertion, given a close to full hash table?