Name: ________________________________

CS 24
Final Exam
Summer 2013 C

Recursion	/18
Sorting	/16
Binary Search Trees	/8
Priority Queues	/8
Hash Tables	/8
Total	**/58**

Please do not begin until told to do so.

Exam Rules

1. No one may leave for any reason and come back to work on the test. If you need to use the bathroom, do it now.
2. There may not be any extra material on your desk or adjacent ones.
3. You may not wear any sunglasses, hats, or hoods.
4. You may not use any electronic devices of any kind. Any headphones need to be put away.
5. Please don't sit in the front row of the classroom.
6. If you appear to be looking around, you will be moved to the front row.

Recursion

```
int bar(Node<int> *n, int c) {  // problems 1 and 2
  while (n != NULL) {
    if (n->data == c)
      return 1;
    n = n->next;
  }
  return 0;
}
int foo(BinaryNode<int> *n, int c) {  // problems 3, 4, and 5
  if (n == NULL)
    return 0;
  int t;
  if (n->data > c)
    t = 1;
  else
    t = 0;
  return t + foo(n->lhs, c) + foo(n->rhs, c);
}
```

1. (3 pts.) What is the purpose of the above **bar** function?

2. (5 pts.) Rewrite **bar** as a recursive function.

3. (3 pts.) What is the purpose of the **foo** function on the previous page?

4. (2 pts.) If the *root* of a *complete tree* containing 4 nodes is passed in during the initial call to **foo** (as given), what is the maximum number of activation records created on the stack at any given time? The initial call to **foo** with the tree root counts as the first activation record on the stack.

5. (5 pts.) Rewrite **foo** as an iterative function.

Sorting

6. (4 pts.) Show the result of each round of selection sort of the following items. Assume the method of selection uses swapping rather than removal + insertion.

3 1 5 8 2 7

7. (4 pts.) Show the result of each round of insertion sort of the following items.

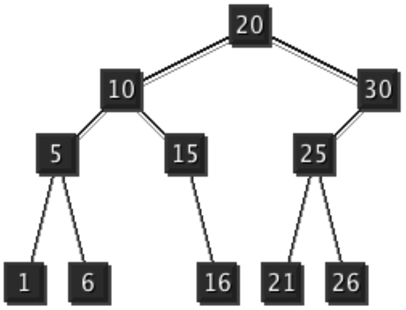
8 7 5 3 2 1

Binary Search Trees

8. (4 pts.) Draw the binary search tree after inserting the following items.

```
15   48   7   -1   29   31   11
```

9. (4 pts.) Redraw the following binary search tree after removing 25, 10, and 20. (Recommended: draw the tree after each removal)



Priority Queues

10. (4 pts.) Draw the contents of the array that represents a **min-heap** after calling **enqueue** on each of the following items. The heap is initially empty.

```
15  48  7  -1  29  8
```

11. (4 pts.) Draw the contents of the array that represents the following **max-heap** after calling **dequeue** 3 times. The following are the initial contents of the array.

```
60  40  10  25  0  3  1  4
```

Hash Tables

12. (4 pts.) Draw the contents of a **hash table** of size 8 with a linear probing constant of 3 after inserting the following items. Hash function: `h(x) = x % 8`

```
7   2   15   5   12   4
```

13. (4 pts.) Assume we have a poorly implemented **hash table** of size 8 with a linear probing constant of 1 that does not properly record deletions. Beginning with the following filled hash table, which elements (numbers) are no longer *accessible* after removing both 10 and 14? Hash function: `h(x) = x % 8`

```
|  8  |  7  |  10  |  1  |  12  |  2  |  14  |  6  |
```