

CS200 Quiz 1: Pre-Requisites (8/28/08,9/2/08)

Key (answers in blue italics, points in red sans serif)

1. The implementation of a linked list data structure in Java usually takes two classes. What are the two classes and what role do they serve? [20]
 1. *A linked list class which includes at least an instance variable for the head of the list*
 2. *A node class which includes at least instance variables for the next node in the list and the data to be kept in this node*
2. Writing an inductive proof requires at least two cases/parts. What are they? [20]
 1. *base case*
 2. *inductive or repeating case*
3. What is the invariant for insertion sort? (If you are not sure how to answer this, then alternatively, what does insertion sort do to the array on each pass?) [15]

Part of the array (up to the number of passes done) will be completely sorted OR it inserts the next unsorted number into its correct position within the sorted portion of the array and increments sorted index of array.
4. How many strings of length 5, 6, or 7, containing letters (52) and digits (10) are there? [15]

$62^5 + 62^6 + 62^7$
5. Which is more time efficient in the worst case: a) the add method for an ArrayList or b) the add (to end) method for a singly linked list? [15]

In standard implementations, they are the same $O(n)$ because both add to the end. In worst case, ArrayList needs to recopy entire array to expand space. If SLL includes a tail instance variable or one assumes adding to the front, then SLL is more efficient.
6. Given this code:

```
public static int mystery (int myArray[], int abc, int def, int ghi) {
    int jkl;
    if (abc > def) { jkl = -1; }
    else {
        int mno = (abc + def)/2;
        if (ghi == myArray[mno]) { jkl = mno; }
        else if (ghi < myArray[mno]) {
            jkl = mystery(myArray, abc, mno-1, ghi);
        }
        else {
            jkl = mystery(myArray, mno+1, def, ghi);
        }
    }
    return jkl;
}
```

 - a. How many times will `mystery` be called given the following two lines:
`int [] anArray = {2,12,17,32,36,87,222};`
`int answer = mystery(anArray, 0, 6, 2)?` [3]

4 or 3 if you don't count the initial call
 - b. What will the value of `answer` be at the end of execution? [3]

0
 - c. What does `mystery` do in general? [Do not simply paraphrase the lines of code in English. Hint: it is an algorithm you should have seen before.] [9]

It's the binary search algorithm.