CS4321 Homework 1

Due on Thursday, Sept. 20 at the beginning of class.

Total 109 points

1. (24 points) Consider the following bubble sort algorithm.

```
void bubblesort(int a[], int n)
{
  boolean noExchg;
  int i, j, tmp;

  for (i=0; i<n-1; i++) {   // outer loop
    noExchg = true;

    for (j=0; j<n-i-1; j++) { // inner loop
      if (a[j] > a[j+1]) {
        tmp = a[j];
        a[j] = a[j+1];
        a[j+1] = tmp;
        noExchg = false;
      }
    }
    if (noExchg)
      return;
  }

}
```

   (a) (8 points) Show that at the end of first loop iteration of the outer loop, a[n-1] = max(a[0..n-1]). (Hint: find a loop invariant for the inner loop and prove it).

   (b) (8 points) What is the best case of this algorithm? Calculate the best-case cost of the algorithm following the style we used in class for insertion sort. Is your function in $O(n^2)$? How about $O(n)$?

   (c) (8 points) Repeat (b) for the worst case.

2. (12 points) Problem 1-1, page 13. You only need to do function lgn, n^2 and 2^n for columns, 1 second, 1 day, and 1 century. For simplicity, assume that each year has 365 days. What do you learn from the table you generated?

3. (10 points) Problem 2.1-3, page 21.

4. (10 points) Problem 2.3-3, page 36. This problem is for practicing mathematical induction. You don't need to read Section 2.3 which we will cover later this semester.

5. (10 points) Problem 3.1-1, page 50.

6. (10 points) Problem 3-1, page 57.

7. (15 points) Problem 3-2 a, b d, page 58. Note that you only need to do rows a, b and d, and just need to answer yes or no.

8. (18 points) For each of the following statements, tell whether it is "true" or "false". Explain why only when you answer "false".

   (a) It is possible that there is an algorithm whose worst case cost is in $\Omega(n^3)$ and in $O(n^2)$.

   (b) It is possible that there is an algorithm whose worst case cost is in $\Omega(n^3)$ and whose best case cost is in $O(n^2)$.

   (c) It is possible that there is an algorithm whose worst case cost is in $\Omega(n \log n)$ and whose best case cost is in $O(n \log n)$.

   (d) It is possible that there is an algorithm whose worst case cost is in $\Omega(n \log n)$ and $O(n \log n)$.

   (e) It is possible that there is an algorithm whose worst case cost is $O(n^2)$ and for some value of n there is at least one instance of size n whose cost is n^3.

   (f) It is possible that there is an algorithm whose worst case cost is $O(n^2)$ but for each value of n, all but one instance of size n have costs bounded by cn for a single constant $c > 0$.