Homework 1
CS 3385
Due Jan 20 in class

1. (25 points) Consider the searching problem:
   Input: A sequence of n numbers $A = (a_1, a_2, \ldots, a_n)$ and a value v.
   Output: An index i such that $v = A[i]$ or the special value `NIL` if v does not appear in A.
   Consider the function `linearSearch`, which scans through the sequence, looking for v.
   (a) Write pseudocode for `linearSearch`.
   (b) Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.
   (c) How many elements of the input sequence need to be checked on the average, assuming that the element being searched for is equally likely to be any element in the array?
   (d) How many elements of the input sequence need to be checked in the worst case?
   (e) What are the average-case and worst-case running times of linear search in Θ-notation? Justify your answers.

2. (4 points) Express the function $n^3/1000 - 100n^2 + 3$ in terms of Θ-notation.

3. (20 points) Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A.
   (a) Write pseudocode for this algorithm.
   (b) What loop invariant does this algorithm maintain?
   (c) Why does it need to run for only the first $n-1$ elements, rather than for all n elements?
   (d) Give the best-case and worst-case running times of selection sort in Θ-notation. Justify your answer.

4. (15 points) We can express insertion sort as a recursive procedure as follows. In order to sort $A[1..n]$, we recursively sort $A[1..n-1]$ and then insert $A[n]$ into the sorted array $A[1..n-1]$.
   (a) Write pseudocode for a recursive insertion sort.
   (b) Write a recurrence for the worst-case running time of this recursive insertion sort.

5. (12 points) Suppose we wish to find the index of a value v in an array A (the searching problem). Observe that if the sequence A is sorted, we can check the midpoint of the sequence against v and eliminate half of the sequence from further consideration. The binary search algorithm repeats this procedure, halving the size of the remaining portion of the sequence each time.
   (a) Write recursive pseudocode for binary search.
   (b) Argue that the worst-case running time of binary search is $\Theta(\log n)$.

6. (8 points) Let $f(n)$ and $g(n)$ by asymptotically nonnegative functions. Prove that
   $$O(f(n) + g(n)) = \max(O(f(n)), O(g(n)))$$
   Hint: One approach is to prove it using two cases, $O(f(n)) < O(g(n))$ and $O(f(n)) > O(g(n))$.

7. (8 points) Show that for any real constants a and b, where $b > 0$,
   $$(n + a)^b = \Theta(n^b)$$

8. (4 points) Is $2^{n+1} = O(2^n)$? Justify your answer.

9. (4 points) Is $2^{2n} = O(2^n)$? Justify your answer.