Analysis of Algorithms
Midterm Examination
Duration: 1 hour

Name: ___________________________ Student Number: ___________________________

1. (a) Let \( f(n) \) and \( g(n) \) be asymptotically non-negative functions. Prove that \( \Theta(f(n)+g(n)) = \max(f(n),g(n)) \).

(b) Prove (formally) that the worst-case running time of quicksort is \( \Theta(n^2) \).

(c) Briefly describe what we mean by a randomized algorithm. Why are we using randomization in quicksort?

2. Prove or disprove the following conjecture: \( f(n) = O(g(n)) \) implies \( 2^{f(n)} = O(2^{g(n)}) \).

3. (a) Determine an asymptotic bound for the following recurrence: \( T(n) = T(n-1) + n \).

(b) Use recursion trees to determine a tight asymptotic bound for the following recursion: \( T(n) = T(n/10) + T(9n/10) + n \).

4. The following "elegant" sorting algorithm has been proposed:

```
STOOGE-SORT(A,i,j)
if A[i] > A[j] then
if i+1 \geq j then return /* returns nothing ... */
k <-- \lfloor (j - i + 1)/3 \rfloor /* round down */
STOOGESORT(A,i,j-k) /* first two-thirds */
STOOGESORT(A,i+k,j) /* last two-thirds */
STOOGESORT(A,i,j-k) /* first two-thirds again */
```

(a) Give a recurrence for the worst-case running time of STOOGE-SORT.

(b) Give a tight asymptotic (\( \Theta \)-notation) bound on the worst-case running time by solving the recurrence.

5. Describe an efficient algorithm that, given \( n \) integers in the range of 1 to \( k \), preprocess the input and then answers any query about how many of the \( n \) integers fall into the range \([a..b]\) in \( O(1) \) time.