This homework will not be graded and it will not count as one of the graded assignments. Completion, however, is mandatory. Please answer the questions in simple, concise prose and when appropriate simple, concise pseudocode. Please do not use any outside resources, including your textbook, when answering these questions. Don't worry if you are uncertain about an answer or if you do not know an answer or if something looks unfamiliar. Just be honest. It will be fine. I promise.

- Seriously, what do you think of when you see the words *algorithm design and analysis*?
- What was your favorite part of CS 134 or CS 136? Why?
- For each of the following, answer with the tightest upper bound from this list : $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$.
  - number of leaves in a depth-n balanced binary tree
  - depth of an n-node balanced binary tree
  - number of edges in an n-node tree
  - worst-case run time to sort n items using merge sort
  - number of distinct subsets of a set of n items
  - number of bits needed to represent the number n
  - time to find the closest pair of points among n points in Euclidean space by enumeration
  - time to insert n items into a binary heap
  - time to find the second largest number in a set of n (not necessarily sorted) numbers
- How many times can you repeatedly halve 32 *before* falling below 1? What is the common mathematical name for this operation?
- We will not use the syntax of any particular programming language in this course to specify algorithms. Instead, we will use well-written prose and pseudocode—an intuitive set of instructions that are appropriate to the problem at hand. For example, Algorithm 1 below gives some pseudocode for finding the smallest integer in an array.

Algorithm 1 FINDMIN(A, n)

Require: An array of integers A of length $n \geq 0$. Empty arrays return $+\infty$.

```
1: m ← +∞
2: for i ← 1 to n do
3:     m ← min(m, A[i])
4: end for
5: return m
```

Write some pseudocode to find the smallest integer value in a binary tree T with left child $T.left$ and right child $T.right$ and integer value $T.value$. You can suppose that $T.left$ (respectively $T.right$) is NULL if it doesn't have a left (respectively right) child.

- Do you have any questions for me? If so, what are they?
- I am reading *Middlesex*, listening to some Bad Bad Hats, and watching Homeland. Do you have any book, music, or television recommendations for me?