CS41 Homework 3

This homework is due at 11:59PM on Wednesday, September 21. Write your solution using LaTeX. Submit this homework in a file named `hw3.tex` using **github**. This is an individual homework. It's ok to discuss approaches at a high level. In fact, we encourage you to discuss general strategies. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner *while in lab*. In this case, note (in your **README file**) who you've worked with and what parts were solved during lab.

When you submit homework assignments this semester, please keep the following in mind:

- Don't forget to fill out the README.md file.
- Don't include your name in hw3.tex. (I'd like the graders to *not* know who you are, to minimize grader bias)
- Don't submit a .pdf – just the .tex will do.
- Graders will compile your code from the .tex file using pdflatex. It is your responsiblity to make sure the LaTeX compiles.

The main **learning goals** of this homework assignment are to practice algorithmic analysis skills.

1. **Asymptotic analysis.** Arrange the following functions in ascending order of growth rate. That is, if g follows f in your list, then it should be the case that $f = O(g)$.

   - $f_1(n) = \frac{\sqrt{n}}{6}$
   - $f_2(n) = 12n\log(n)$
   - $f_3(n) = 5\log(n)^4$
   - $f_4(n) = \pi \cdot 2^n$
   - $f_5(n) = 7n^3$
   - $f_6(n) = 16n^2 + 22n$

   No proofs are necessary.

2. **Analysis.** Let $f(n) = 99n^{2.5}$ and $g(n) = n^2(\log n)^8$. Prove that $g(n) = O(f(n))$. You may use techniques and facts from class and the textbook; your proof should be formal and complete.

3. **Close to sorted.** Say that a list of numbers is "k-close-to-sorted" if each number in the list is less than k positions from its actual place in the sorted order. (So a 1-close-to-sorted list is *actually* sorted.) Give an $O(n\log k)$ algorithm for sorting a list of numbers that is k-close-to-sorted.

   In your algorithm, you may use any data structure or algorithm from CS35 by name, without describing how it works.