CSE 421
Introduction to Algorithms
Assignment #1 (rev c)
Due: Friday, 4/7/17

Instructions:

• Homeworks generally are due on Fridays, at the start of class. See the course home page for our late policy.
• Assignments must be turned in electronically; see the FAQ page for instructions.

Important Notes:

• All quarter long, when a problem says “give an algorithm,” unless explicitly stated otherwise, it always means algorithm (English overview plus high-level pseudo-code, as efficient as you can manage), correctness argument and run time analysis. See the FAQ page for more on this, including the level of detail expected.
• For all homework assignments this quarter, it is a violation of my academic integrity policy for you to search for, read, or use solutions to these or similar problems written by others. You may discuss these problems with other students in this class, but you must write your solutions on your own.

Problems:

1. [10 points] Prove by induction that, for all integer \( n > 1 \), \( \frac{1}{n + 1} + \frac{1}{n + 2} + \ldots + \frac{1}{2n} > \frac{13}{24} \).

2. [10 points] If \( p(n) \) is a degree \( d \) polynomial whose high-order term has a positive coefficient, prove that \( p(n) = \Omega(n^d) \). (Hint: see “analysis” slides, circa #22.)

3. [10 points] Given an undirected graph \( G = (V, E) \), a matching in \( G \) is a subset \( M \) of edges, no two of which have a vertex in common; a perfect matching is a matching that includes all vertices. Give an inductive construction of a family of graphs \( \{G_n \mid n \geq 1\} \), with \( 2n \) nodes and \( n^2 \) edges such that \( G_n \) has exactly one perfect matching. Prove, by induction, that your construction is correct.

4. [10 points] Show that \( 2n = o(n!) \). Possibly useful fact: if \( 0 \leq a(n) \leq b(n) \) and \( \lim_{n \to \infty} b(n) = 0 \), then \( \lim_{n \to \infty} a(n) = 0 \).

5. [10 points] Show \( \sum_{i=1}^{n} i^k \log_2 i = \Theta(n^{k+1} \log n) \).

6. [10 points] Suppose functions \( f(n) > 0 \) and \( g(n) > 0 \) satisfy \( f(n) = O(g(n)) \). Prove or give a counterexample to each of the following.
   - (a) \( \log_2 f(n) = O(\log_2 g(n)) \)
   - (b) \( 2^{f(n)} = O(2^{g(n)}) \)
   - (c) \( (f(n))^2 = O((g(n))^2) \)

7. [10 points] Arrange the following six function in order of non-decreasing growth rate; i.e., \( f_i \) may precede \( f_j \) only if \( f_i = O(f_j) \).
   - (a) \( f_1(n) = n^{2.5} \)
   - (b) \( f_2(n) = \sqrt{2n} \)
   - (c) \( f_3(n) = n + 10 \)
   - (d) \( f_4(n) = 10^n \)
   - (e) \( f_5(n) = 100^n \)
   - (f) \( f_6(n) = n^2 \log_2 n \)

Extra Credit: say, and prove, which of the relations are actually \( \Theta \) or little-o, rather than just big-O.
8. [10 points] KT Chapter 2, problem 8a, page 69. (jpg image) **Extra Credit:** 8b.

9. [10 points] KT Chapter 1, problem 4, page 23. (jpg image) You may omit a runtime analysis, but write a paragraph explaining why your algorithm is correct. (This doesn’t have to be very formal, but do try to make it convincing.)

10. [10 points] Assuming that your computer can perform 10 billion operations per second, what is the largest value of $n$ such that you can complete the following number of operation in one hour?

(a) $n^2$
(b) $n^3$
(c) $100n^2$
(d) $n \log_2 n$
(e) $2^n$
(f) $2^{2^n}$