CS180 Homework 1

Due: February 1, 11:59pm

1. (15 pt) Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

   - $f_1(n) = \frac{n^4}{\log n}$
   - $f_2(n) = \sqrt{2n}$
   - $f_3(n) = n^2/2$
   - $f_4(n) = 2^{3\log n}$
   - $f_5(n) = n(\log n)^{100}$
   - $f_6(n) = 2^{\log n - \log\log n}$

2. (18 pt) Let $f(n)$ and $g(n)$ be positive functions (for any n they give positive values) and $f(n) = O(g(n))$. Prove or disprove each of the following statements:

   (a) (6 pt) $g(n) = \Omega(f(n))$

   (b) (6 pt) $f(n) \cdot g(n) = O(g(n)^2)$

   (c) (6 pt) $2^{f(n)} = O(2^{g(n)})$

3. (22 pt) In the class we showed that the number of iterations (number of proposals) in the Gale-Shapley algorithm is upper bounded by $\leq n^2$ for n men and n women, since each man can only make $\leq n$ proposals. However, we haven't shown a lower bound on number of iterations. To show the lower bound is also in the order of n^2, please give a way to construct the preference lists for n men and n women such that the Gale-Shapley algorithm will run for $\Theta(n^2)$ iterations. (For simplicity, you can assume that the algorithm always chooses the unmatched man with the smallest index at each iteration).

4. (20 pt) Let m_1, m_2 be two of the men and w_1, w_2 be two of the women in an instance of the Stable Matching Problem with n men and n women. Assume m_1, m_2, w_1, w_2's preference lists are:

   $$\begin{aligned} m_1\text{'s preference: } & w_1 > w_2 > \dots \\ m_2\text{'s preference: } & w_2 > w_1 > \dots \\ w_1\text{'s preference: } & m_2 > m_1 > \dots \\ w_2\text{'s preference: } & m_1 > m_2 > \dots \end{aligned}$$

   So, we only know the favorite and the second favorite of each of these four persons.

   (a) (10 pt) Show that if we run Gale-Shapley algorithm with men proposing, then m_1 is matched to w_1 and m_2 is matched to w_2.

   (b) (10 pt) Show that in **every** stable matching, m_1, m_2 are matched to w_1, w_2, i.e., $(m_1, w_1), (m_2, w_2)$ or $(m_1, w_2), (m_2, w_1)$ must be part of any stable matching.

5. (25 pt) Given an array A of n distinct integers and assume they are sorted in increasing order. Design an algorithm to find whether there is an index i with $A[i] = i$. The algorithm **should** run in $O(\log n)$ time. For the answer, please briefly write down your key observations of this problem, provide the pseudo codes with comments, and analyze your time complex. Hint: recursive function.

★ Homework assignments are due on the exact time indicated. Please submit your homework using the Gradescope system. Email attachments or other electronic delivery methods are not acceptable. To learn how to use Gradescope, you can:

- 1. Watch the one-minute video with complete instructions from here: https://www.youtube.com/watch?v=-wemznvGPfg
- 2. Follow the instructions to generate a PDF scan of the assignments: http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf
- 3. **Make sure you start each problem on a new page.**

★ We recommend to use LaTeX, LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receiver fewer points.

★ Unless specified, you should justify the correctness and time complexity of your algorithm.

★ **Your answer to each problem should be written in 2 (A4 or letter sized) pages.** If you are typing for the homework, a font size of 12 points or larger must be used. If you are writing, the font size should not be too small to affect the readability. You will get 5 points penalty for each additional page used. For all these problems, the correct solution can be easily written in 1 page.