CSE421: Design and Analysis of Algorithms	January 11, 2018
Homework 2	
Shayan Oveis Gharan	Due: January 18, 2018 at 5:00 PM

Please see https://courses.cs.washington.edu/courses/cse421/18wi/grading.html for general guidelines about Homework problems.

Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers.

P1) Arrange the following in increasing order of asymptotic growth rate. For full credit it is enough to just give the order.

(a) $f_1(n) = \frac{n^4}{\log n}$

(b) $f_2(n) = n^2/2$

(c) $f_3(n) = 2^{3\sqrt{\log n}}$

(d) $f_4(n) = n(\log n)^{1000}$

(e) $f_5(n) = 2^{n \log n}$

(f) $f_6(n) = 2^{\log n - \log\log n}$

P2) In this problem we want to design another algorithm for largest sum-subsequence problem that also runs in linear time. The idea is to use the *preprocessing* technique. Remember that in the preprocessing technique we construct a data structure and we use it to answer queries on the input data efficiently.

a) Let $x_1, \ldots, x_n$ be the given input sequence. We construct an array of length $n+1$, called *sum* such that for all i,

$$sum[i] = x_1 + \cdots + x_i.$$

We also assume $sum[0] = 0$. Show that this array can be constructed in linear time.

b) Now we use the above array to solve our problem. Show that for any $i \leq j$

$$x_i + \cdots + x_j = sum[j] - sum[i-1].$$

c) Show that for every i the largest sum subsequence that ends at i is

$$sum[i] - \min_{0 \leq j \leq i} sum[j].$$

d) Design a linear time algorithm that outputs the sum of the numbers in the consecutive subsequence with the largest sum.

P3) Let $x_1, \ldots, x_n$ be a sequence of integers (not necessarily positive) given in the input. Use induction to design an $O(n)$-time algorithm to find a subsequence $x_i, x_{i+1}, \ldots, x_j$ of consecutive elements such that the product of elements in it is maximum over all consecutive subsequences.

P4) Suppose n persons are at a party. Suppose that friendship is a symmetric relationship, that is i is a friend of j if and only if j is a friend of i. Prove that two of the individuals at the party have exactly the same number of friends.

Hint: Model this problem as a graph question, and argue that if all vertices of a graph have distinct degrees, then the vertices must have degrees $0, 1, 2, \ldots, n-1$, and this cannot happen.

P5) (**Extra Credit**) Let f be a function that maps A to A. We say a set $S \subseteq A$ is bijective if f maps S to S and f is one-to-one on S. In class we designed an algorithm that finds the largest bijective subset of A. Design a *linear time* algorithm (i.e., $O(n)$) that returns the number of bijective subsets of A. For example, if $A = \{1, 2\}$ and $f(1) = 1$ and $f(2) = 2$ your algorithm must return 4 because every subset of A is bijective.