In this and all future homeworks, please clearly print the day and meeting time of *one* discussion section at the top of each page, together with the homework number, problem number, your names, and your NetIDs. We will return your graded homeworks at the discussion section you indicate. For example:

CS 473 Homework 3 Problem 2

Kyle Jao (fjao2), Nirman Kumar (nkumar5), Ernie Pan (erniepan)
Wednesday 3pm section

1. You are given a set A of n arcs on the unit circle, each specified by the angular coordinates of its endpoints. Describe an efficient algorithm for finding the largest possible subset X of A such that no two arcs in X intersect. Assume that all $2n$ endpoints are distinct.
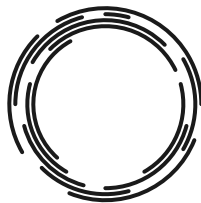
2. Suppose you are given an $m \times n$ bitmap, represented by an array $M[1..m, 1..n]$ whose entries are all 0 or 1. A *checkered block* is a subarray of the form $M[i..i', j..j']$ in which no pair of adjacent entries is equal. Describe an efficient algorithm for finding the number of elements in the largest checkered block(s) in M.

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Two checkered blocks in a 6×6 bitmap

3. A company is planning a party for its employees. The employees in the company are organized in a strict hierarchy, that is, a tree with the company president at the root. The organizers of the party have assigned a real number to each employee indicating the *awkwardness* of inviting both that employee and their immediate supervisor; a negative value indicates that the employee and their supervisor actually like each other. The organizers want to selectively invite employees to the party so that the the total awkwardness is as small as possible. For example, if the guest list does not include both an employee and their supervisor, the total awkwardness is zero.

   However, the president of the company insists on inviting exactly k employees to the party, including herself. Moreover, everyone who is invited is required to attend. Yeah, that'll be fun.

   Describe an algorithm that computes the total awkwardness of the least awkward subset of k employees to invite to the party. The input to your algorithm is an n-node rooted tree T representing the company hierarchy, an integer $awk(x)$ for each node x in T, and an integer $0 \le k \le n$. Your algorithm should return a single integer.

   For full credit, you may assume that the input is a <u>binary</u> tree. A complete solution for <u>arbitrary</u> rooted trees is worth 10 points extra credit.

4. [**Extra credit**] Two players A and B play a turn-based game on a rooted tree T. Each node v is labeled with a real number $\ell(v)$, which could be positive, negative, or zero.

   The game starts with three tokens at the root of T. In each turn, the current player moves one of the tokens from its current node down to one of its children, and the current player's score is increased by $\ell(u) \cdot \ell(v)$, where u and v are the locations of the two tokens that did *not* move. At most one token can be on any node (except the root) at any time. The game ends when the current player is unable to move, for example, if all three tokens are at leaves. The player with the higher score at the end of the game is the winner.

   Assuming that both players play optimally, describe an efficient algorithm to determine who wins on a given labeled tree. Do not assume that T is a *binary* tree.