1. **2D pattern matching.** Suppose that $X$ is an $n \times n \{0, 1\}$-matrix and $Y$ is an $m \times m \{0, 1\}$-matrix for $m < n$. One wants to check if $Y$ occurs as a block submatrix of $X$. Describe how to change the Karp-Rabin algorithm for 1D pattern matching to handle this. Your algorithm should run in $O(n^2)$ time. You may assume that arithmetic on $O(\log n)$-bit integers can be done in constant time. Make sure you justify why the the integers arising in your algorithm are of this size. You may also assume (as we did in lecture) that it’s possible to choose a uniformly random prime $p$ from a given interval.

[Note that you will have to use some thought to get a running time of $O(n^2)$, since it’s no longer clear how to update the hashes at every step.]

2. **The second moment method.** Consider the following simple model for a social network: There are $2n$ users in two groups $A$ and $B$ with $|A| = |B| = n$. For every distinct pair of users $i$ and $j$, the pair $\{i, j\}$ are friends independently with probability $p$ if they are in the same group, and probability $q$ if they are in different groups. A **loner** is a user $i$ that has no friends.

Let $L$ denote the event that there exists a loner. Prove that if $p + q \gg \frac{\ln n}{n}$, then $\mathbb{P}[L] \to 0$ and if $p + q \ll \frac{\ln n}{n}$, then $\mathbb{P}[L] \to 1$. [Here the notation $f(n) \gg g(n)$ means that $\lim f(n)/g(n) = \infty$.]

3. **The probabilistic method.** Suppose that a group of $n$ senators gather for a press conference. Each senator is either a democrat or a republican and chooses to wear one colored pin to show support for a cause they “believe in.”

It is considered a major political faux pas if two members of different parties wear pins of the same color. Show that if every senator has more than $\log_2 n$ causes they support (each with a different color), then it is possible for them to select pins so that no pair of senators from different parties wear the same color.

[For clarification: Each person has a list of $> \log_2 n$ colors. The goal is to show that every person can choose a color from their list so that no two people from different parties have the same color.]

**Bonus:** Can you give a polynomial-time deterministic algorithm to assign a color to every senator?