# Homework 7

Out: 20 Mar. Due: 5 Apr.

*Instructions: Put your solutions in the homework box on Soda level 2 by 5pm on Thursday. Take time to write* clear *and* concise *answers; confused and long-winded solutions will be penalized. You are encouraged to form small groups (two to four people) to work through the homework, but you* **must** *write up all your solutions on your own. Depending on grading resources, we reserve the right to grade a random subset of the problems and check off the rest; so you are advised to attempt all problems.*

**1.** MU, Exercise 6.9.

**2.** MU, Exercise 6.4. [HINT: In part (b), fix a probability distribution over the remover's strategies, and compute the expected number of tokens that reach position $n$. In particular, you will need to compute, for a fixed token, the probability that the token reaches position $n$. For the appropriate distribution, this quantity is (somewhat surprisingly) independent of the chooser's strategy.]

**3.** In this problem we will see that the value $p = \frac{\ln n}{n}$ is a "threshold" for the property that a random graph in the $\mathcal{G}_{n,p}$ model has an *isolated vertex*, i.e., a vertex with no adjacent edges. That is, we will prove that

$$\Pr[G \text{ has an isolated vertex}] \overset{n\to\infty}{\longrightarrow} \begin{cases} 0 & \text{if } p = \omega(\frac{\ln n}{n}); \\ 1 & \text{if } p = o(\frac{\ln n}{n}). \end{cases}$$

(a) Let the r.v. $X$ denote the number of isolated vertices in $G$. Write down the expectation of $X$ as a function of $n$ and $p$.

(b) Show that $E[X] \to 0$ for $p = \omega(\frac{\ln n}{n})$, and that $E[X] \to \infty$ for $p = o(\frac{\ln n}{n})$.

(c) Deduce from part (b) that $\Pr[G \text{ has an isolated vertex}] \to 0$ for $p = \omega(\frac{\ln n}{n})$.

(d) Compute $\mathrm{Var}[X]$ as a function of $n$ and $p$.

(e) Deduce from parts (b) and (d) that $\Pr[G \text{ has an isolated vertex}] \to 1$ for $p = o(\frac{\ln n}{n})$.

**4.** We consider a problem motivated by recommendation systems used by online merchants such as Amazon and Netflix. Given two sets of integers $A, B$ of size $n$, we would like to quickly determine if $A = B$, or if $|A \cap B|$ is very small, say $|A \cap B| < 0.01n$. (In the intermediate case, where $A \cap B$ is of moderate size, we do not care what the output is.) In the case of Amazon's recommendation system, $A$ and $B$ could be the lists of books purchased by two different customers, and $n$ could be very large, e.g. $1$ billion.

(a) Sketch a simple deterministic algorithm that computes $|A \cap B|$ exactly using $O(n \log n)$ comparisons.

Our aim is to beat this algorithm, using randomization and exploiting the fact that we only want to distinguish the case where $A = B$ from the case where they are very different. Specifically, we seek an algorithm with the following properties:

— if $A = B$, then the algorithm outputs "yes" with probability at least $3/4$.

— if $|A \cap B| \le 0.01n$, then the algorithm outputs "no" with probability at least $3/4$.

— the algorithm uses only $O(\sqrt{n} \log n)$ comparisons.

[The value 3/4 here is for convenience only; as we have seen in class, it can easily be boosted to $1 - \delta$ for any desired $\delta > 0$ using only $O(\log(1/\delta))$ repeated trials.]

Here is a proposed algorithm, where $c$ is a constant to be determined:

(1) choose a subset $X$ of $A$ by picking each element of $A$ independently with probability $c/\sqrt{n}$

(2) choose a subset $Y$ of $B$ by picking each element of $B$ independently with probability $c/\sqrt{n}$

(3) if $|X| > 2c\sqrt{n}$ or $|Y| > 2c\sqrt{n}$, output "yes";

(4) compute $|X \cap Y|$; if $|X \cap Y| \geq 0.1c^2$, output "yes"; else, output "no".

In the rest of this problem, we will show that the algorithm achieves the required properties for a sufficiently large constant $k$.

(b) Show that the algorithm does indeed use only $O(\sqrt{n} \log n)$ comparisons, assuming that $c$ is a constant.

(c) Suppose $A = B$. Show that the algorithm outputs "yes" with probability at least $1 - e^{-0.81c^2/2}$.

(d) Suppose $|A \cap B| \leq 0.01n$. Show that the algorithm outputs "yes" with probability at most $e^{-0.81c^2/11} + 2e^{-\Omega(\sqrt{n})}$.

(e) Indicate briefly how to choose the constant $c$ so as to achieve the 1/4 error probabilities specified earlier. (You need not actually perform this calculation.)