

Due: Friday 1/17, 11:59pm on Gradescope

Please follow the homework policies on the course website.

1. (5 pt.) [Perfect Matchings.]

Let  $G = (V, E)$  be a *bipartite graph* with  $n$  vertices on each side. A *perfect matching* in  $G$  is a list of edges  $M \subset E$  so that every vertex in  $V$  is incident to exactly one edge.

For example, here is a bipartite graph  $G$  (on the left), and a perfect matching in  $G$  (shown in **bold** on the right):



Your goal is to determine if the graph  $G$  has a perfect matching.

There are efficient deterministic algorithms for this problem, but in this problem you'll work out a simple randomized one.<sup>1</sup>

- (a) (2 pt.) Recall that the *determinant* of an  $n \times n$  matrix  $A$  is given by

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n A_{i, \sigma(i)},$$

where the sum is over all permutations  $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ , and where  $\text{sgn}(\sigma)$  denotes the signature<sup>2</sup> of the permutation  $\sigma$ . (For example, if  $n = 3$ , then the function  $\sigma : \{1, 2, 3\} \rightarrow \{1, 2, 3\}$  that maps  $1 \mapsto 2$ ,  $2 \mapsto 1$ ,  $3 \mapsto 3$  is a permutation in  $S_3$ . The signature of  $\sigma$  happens to be  $-1$ , although as noted in the footnote, if you haven't seen this definition before, don't worry about it).

Let  $A$  be the  $n \times n$  matrix so that

$$A_{ij} = \begin{cases} x_{ij} & (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

where the  $x_{ij}$  are variables, and  $(i, j) \in E$  if and only if the  $i$ -th vertex on the left and the  $j$ -th vertex on the right are connected by an edge in  $G$ . Notice that  $\det(A)$  is a multivariate polynomial in the variables  $x_{ij}$ .

Explain why  $\det(A)$  is not identically zero if and only if  $G$  has a perfect matching.

<sup>1</sup>This randomized algorithm has the advantage that (a) it generalizes to all graphs (not necessarily bipartite), and (b) it can be parallelized easily. Moreover, it's possible to generalize it to actually recover the perfect matching (and not just decide if there is one or not).

<sup>2</sup>The *signature* of a permutation is defined as  $-1$  if the permutation can be written as an odd number of transpositions, and  $+1$  otherwise. **The exact definition isn't important** to this problem, all you need to know is that it's either  $\pm 1$  in a way that depends on  $\sigma$ .

- (b) **(3 pt.)** Use the part above to develop a randomized algorithm for deciding whether or not there is a perfect matching. Your algorithm should run in  $O(n^3)$  operations. If  $G$  has no perfect matching, your algorithm should return “There is no perfect matching” with probability 1. If  $G$  has a perfect matching, your algorithm should return “There is a perfect matching” with probability at least 0.9.

You should clearly state your algorithm and explain why it has the desired properties.

[**HINT:** *You may use the fact that one can compute the determinant of a matrix  $A \in \mathbb{R}^{n \times n}$  in  $O(n^3)$  operations. ]*

- (c) **(0 pt.)** [**Optional: this won't be graded.**] Extend your algorithm to actually *return* a perfect matching. And/or, extend your algorithm to non-bipartite graphs. As a hint, consider the matrix

$$A = \begin{cases} x_{ij} & \{i, j\} \in E \text{ and } i < j \\ -x_{ji} & \{i, j\} \in E \text{ and } i \geq j \\ 0 & \text{else} \end{cases}$$

2. **(6 pt.)** [**Coin Flipping**] Suppose you are flipping a fair coin repeatedly. (Here, a *fair coin* means that it comes up heads with probability 1/2 and tails with probability 1/2; each flip is independent).

- (a) **(3 pt.)** What is the expected number of flips until you get two heads in a row (counting both heads)? Justify your answer.

[**HINT:** *If you find yourself doing a tedious computation, try to think of a simpler way. Perhaps look to the mini-lecture on linearity of expectation for some inspiration... ]*

- (b) **(3 pt.)** What is the expected number of flips until you get  $k$  heads in a row (counting all  $k$  heads)? Justify your answer.

[**HINT:** *You may want to do it for  $k = 3$  first to get some intuition. ]*

[**HINT:** *Depending on how you approach it, the following facts might be helpful:  $\sum_{j=1}^t \frac{j}{2^j} = \frac{1}{2^t}(2^{t+1} - t - 2)$ , and  $\sum_{j=1}^t 2^{-j} = 1 - 2^{-t}$ . ]*

**Note:** *There are many ways to do this problem, but there is a not-too-long way to do it by doing something similar to something we did in the mini-lecture on linearity of expectation.*

3. **(10 pt.)** [**More coin flipping**] Suppose you are given a fair coin and want to use it to “simulate” a coin that lands heads with probability exactly 1/3. Specifically, you will design an algorithm whose only access to randomness is by flipping the fair coin (repeatedly, if desired), and your algorithm should return “heads” with probability exactly 1/3 and “tails” with probability exactly 2/3.

- (a) **(4 pt.)** Prove that it is *impossible* to do this if the algorithm is only allowed to flip the fair coin at most 1,000,000,000 times.

[**HINT:** *Read the next two parts of the problem first... ]*

- (b) **(4 pt.)** Design an algorithm for the above task that flips the fair coin a finite number of times *in expectation*.

- (c) **(2 pt.)** Show that for *any* value  $v$  in the interval  $[0, 1]$ , there is an algorithm that flips a fair coin at most 2 times in expectation, and outputs “heads” with probability  $v$  and “tails” with probability  $1 - v$ .

**Note:** if you do this part correctly, you can write “follows from (c)” in part (b) get full credit for both parts.

[**HINT:** *Think about representing the desired probability in its binary representation.* ]

4. **(10 pt.) [Favorite coffeeshops]** There are  $n$  coffeeshops on the Stanford campus, all of a different quality. You are on a mission to find the best coffeeshop. Your strategy is as follows: you arrange the coffeeshops in a random order, and visit each shop once, in that order, and assess its quality. Let  $q_t$  be the quality of the  $t$ 'th coffeeshop you visit. (That is,  $q_1, \dots, q_n$  is a random permutation of  $n$  distinct numbers).

- (a) **(3 pt.)** Say that a coffeeshop  $t$  is a *new favorite* if it's better than all the coffee shops you have been to before: that is, if  $q_t > q_1, q_2, \dots, q_{t-1}$ . (By definition, the first coffeeshop is a new favorite). If you visit all  $n$  coffeeshops, what is the expected number of new favorites you discover? Give your answer in the form of a summation from 1 to  $n$ .
- (b) **(2 pt.)** Show that your answer to part (a) satisfies

$$\ln(n + 1) \leq [ \text{your answer to part (a)} ] \leq 1 + \ln(n).$$

[**HINT:** *Compare your sum from part (a) to an integral.* ]

- (c) **(3 pt.)** Instead of visiting all of the coffeeshops, you come up with a different strategy, which may allow you to stop earlier. You will visit the first  $k$  coffeeshops to get an idea of the distribution, where  $k < n$ . Then you will stop at your first new favorite *after* those first  $k$ , and declare it to be your favorite for all time. (If you don't find a new favorite by the  $n$ 'th coffeeshop, you won't have a favorite-of-all-time, and presumably you'll never go to a coffeeshop again).

*For example, if  $k = 2$  and  $(q_1, q_2, q_3, q_4, q_5, q_6, q_7) = (5, 2, 3, 7, 4, 1, 6)$  (where bigger quality is better), you'll stop at  $q_4$  (which is indeed the best), since that's your first new favorite after the  $k$ 'th (second) coffee shop.*

Show that the probability of finally stopping at the *best* coffeeshop (and declaring it your favorite of all time) is equal to

$$\frac{k}{n} \sum_{t=k+1}^n \frac{1}{t-1}.$$

- (d) **(2 pt.)** Show that you can pick  $k$  so that the probability of finally stopping at the best coffeeshop is at least  $1/e - o(1)$ . You may use the fact that

$$\frac{k}{n} (\ln n - \ln k) \leq \frac{k}{n} \sum_{t=k+1}^n \frac{1}{t-1}.$$

[You don't have to prove this fact on the homework, although it may be fun to do so on your own; it is similar to part (b).]

[**HINT:** *What  $k$  should you pick to maximize  $\frac{k}{n} (\ln n - \ln k)$ ? ]*