

Due: September 25 (Friday) at 23:59 (Pacific Time)

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.¹

- (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 signature² of the permutation σ . (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 σ 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.

¹This 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).

²The *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 ± 1 in a way that depends on σ .

- (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. **(8 pt.)** Suppose you are rolling a fair, 6-sided die repeatedly.
- (a) **(4 pt.)** What is the expected number of rolls until you get two 3's in a row (counting both 3's)? Justify your answer.
 [**HINT:** *The answer is not 36...*]
 [**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) **(4 pt.)** What is the expected number of rolls until you get a 3 followed by *either* a 3 or a 4 (counting both rolls)? Justify your answer.
- (c) **(0 pt.)** [**Optional: this won't be graded**] What is the expected number of rolls until you get a 3 followed by a 4 (counting both rolls)?

3. **(12 pt.)** A long time ago (before COVID-19...) a queen wanted to hold a grand party. The queen had n bottles of wine set aside for the party. Unfortunately, a wicked fairy had cursed k of the n bottles of wine, so that they would give severe indigestion to anyone who drinks from them. The queen did not want to give her guests indigestion. The court alchemist had devised a test to identify cursed wine: it only took a drop of wine, but the test was expensive. The queen suggested:

Strategy 1: test all n of the bottles! It will be expensive, but worth it for the comfort of my guests!

The alchemist, who was very clever, suggested another approach. The test could work on *mixtures* of wine: given a drop of wine from several bottles, the test will come up positive if any of the wine was cursed. The alchemist suggested:

Strategy 2: divide the n bottles into n/m disjoint groups of size m , at random. (Suppose for convenience that m divides n). Then test each of the groups, using one test per group. If any group tests positive, re-test each bottle in the group to identify the poison bottles.

- (a) **(4 pt.)** In terms of n, m, k , what is the expected number of groups that will test positive in Strategy 2? It's okay to leave your answer in terms of some binomial coefficients or something like that.
- (b) **(2 pt.)** In terms of n, m, k , what is the expected total number of tests that we will need in Strategy 2? Again, it's okay to leave your answer in terms of some binomial coefficients or something like that.
- (c) **(6 pt.)** Consider the following parameter regimes, where we think of $n \rightarrow \infty$. In which of these parameter regimes is Strategy 2 better than Strategy 1, in terms of expected number of tests as $n \rightarrow \infty$?

Below, the “little-oh” notation $f(n) = o(g(n))$ means that $f(n)/g(n) \rightarrow 0$ as $n \rightarrow \infty$; the “little-omega” notation $f(n) = \omega(g(n))$ means that $f(n)/g(n) \rightarrow \infty$ as $n \rightarrow \infty$; and all asymptotic notation is with respect to the limit $n \rightarrow \infty$.

- i. $k = \Theta(1)$, $\omega(1) = m = o(\sqrt{n})$.
- ii. $k = \Theta(n^{5/6})$, $m = \Theta(n^{1/3})$
- iii. $2 \leq m = \Theta(1)$, $k = \Theta(\sqrt{n})$.

[HINT: At this point, you'll probably want to simplify any binomial coefficients you have lying around. There are a number of ways to do this.³ You may want to use the fact (which follows from Stirling's approximation) that for $\omega(1) = k = o(\sqrt{n})$,

$$\binom{n}{k} = (1 + o(1)) \frac{1}{\sqrt{2\pi k}} \left(\frac{ne}{k}\right)^k.$$

It might also help to remember that $e^{-x} = 1 - x + O(x^2)$ as $x \rightarrow 0$. Part of this problem is to get really comfortable with these sorts of approximations!]

- (d) **(0 pt.)**[**Optional: this won't be graded.**] The court jester, who was cleverer still, suggested the following plan, which had the advantage that only one round of testing was needed:

Strategy 3: Create t groups. Put a drop of wine from bottle i into group j with probability p , independently for each (i, j) . (Note that one bottle may appear in several groups). For each bottle, if it is tested negative in at least one group, the bottle is safe and can be served to the guests. Otherwise, declare the bottle cursed.

- i. In terms of p, k, n , what is the expected number of false positives (that is, bottles that are safe but are labeled cursed) in Strategy 3? What is the expected number of false negatives (cursed bottles labeled safe)?
- ii. Suppose that $p = 1/k$ and $k \geq 2$. Say that the queen wants to serve at least $n - 2k$ bottles at the party in expectation, and under no circumstances wants to serve a cursed bottle. How many tests does Strategy 3 use? How does this compare to Strategy 1?

³Here is a delightful—and very useful—note by Shagnik Das for more about various approximations of binomial coefficients: <http://page.mi.fu-berlin.de/shagnik/notes/binomials.pdf>