Philosophy of analyzing randomized algorithms. The first step is to always identify a *bad event*. I.e. identify when your randomness makes your algorithm fail. We will review some techniques from class using the following problem as our "test bed".

Let G be a bipartite graph with n left vertices, and n right vertices on $n^2 - n + 1$ edges.

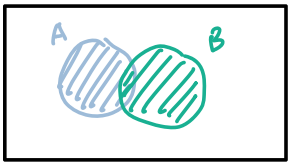
- Prove that G always has a perfect matching.
- Give a polynomial in n time algorithm to find this perfect matching.

We will analyze the following algorithm `BlindMatching`:

- Let $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}$ be independent and uniformly random permutations of $[n]$.
- If $\{\boldsymbol{\pi}(1), \boldsymbol{\sigma}(1)\}, \{\boldsymbol{\pi}(2), \boldsymbol{\sigma}(2)\}, \ldots, \{\boldsymbol{\pi}(n), \boldsymbol{\sigma}(n)\}$ is a valid matching output it.
- Else output `failed`.

Union Bound. Suppose $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are (not necessarily independent) $\{0, 1\}$ valued random variables, then

$$\Pr[\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_n \geq 1] \leq \Pr[\boldsymbol{X}_1 = 1] + \Pr[\boldsymbol{X}_2 = 1] + \cdots + \Pr[\boldsymbol{X}_n = 1].$$



$\mathbb{P}[A \cup B] = \mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[A \cap B]$

$\leq \mathbb{P}[A] + \mathbb{P}[B]$

↑ sometimes greater than 1, often union bound not powerful enough

Now we analyze our algorithm using union bound. An output $M = (\{\boldsymbol{\pi}(1), \boldsymbol{\sigma}(1)\}, \ldots, \{\boldsymbol{\pi}(n), \boldsymbol{\sigma}(n)\})$ is a valid perfect matching exactly when all edges of the form $\{\boldsymbol{\pi}(i), \boldsymbol{\sigma}(i)\}$ are present in G. A "bad event" happens if any of those pairs are not edges in G.

Let $\boldsymbol{X}_i$ be the indicator of the event that $\{\boldsymbol{\pi}(i), \boldsymbol{\sigma}(i)\}$ is *not* present in our graph.

1. What is the probability that $\boldsymbol{X}_i = 1$?

$\mathbb{P}[X_i = 1] = \frac{n-1}{n^2} = \frac{1}{n} - \frac{1}{n^2}$

- n^2 possible pairs
- If there is an edge (u, v), u can be paired with $n-1$ vertices that aren't v

2. Use the union bound to upper bound the probability that M is *not* a valid perfect matching.

$\mathbb{P}[\bigcup_{i=1}^{n} X_i] \leq n\left(\frac{1}{n} - \frac{1}{n^2}\right) = 1 - \frac{1}{n}$

3. Conclude that G has a valid perfect matching.

$\mathbb{P}[M \text{ is perfect matching}] = \frac{1}{n} > 0$

The upper bound obtained on the probability of our bad event, i.e. of M not being a valid perfect matching, is fairly high. In light of this, we introduce the technique of *amplification*.

Amplification. The philosophy of amplification is that if we have a randomized algorithm that fails with probability p, we can repeat the algorithm many times and aggregate the output of all the runs to produce a new output such that the failure probability of the randomized algorithm is significantly smaller. Now consider the following algorithm `SpamBlindMatching`.

- Run `BlindMatching` independently T times.
- If at least one of the runs outputted a valid perfect matching, return the output of such a run.
- Else output `failed`.

with 1 trial:

$\mathbb{P}[\text{fail}] = p \qquad \mathbb{P}[\text{success}] = 1 - p$

with k trials:

$\mathbb{P}[\text{fail all } k] = p^k \qquad \mathbb{P}[\text{success}] = 1 - p^k$

algo successful if ANY one of the trials are successful

1. What is the failure probability of `SpamBlindMatching`?

$\mathbb{P}[\text{fail}] = \left(1 - \frac{1}{n}\right)^T$ note: trials are independent

2. How large should we set T if we want a failure probability of δ?

$$\mathbb{P}[\text{fail}] = \left(1 - \frac{1}{n}\right)^T = \delta$$

$$T \ln\left(1 - \frac{1}{n}\right) = \ln \delta$$

$$T = \frac{\ln \delta}{\ln\left(1 - \frac{1}{n}\right)}$$

Now we switch gears and turn our attention to concentration phenomena and its usefulness in analyzing randomized algorithms.

Markov's inequality. Let $\boldsymbol{X}$ be a *nonnegative valued* random variable, then for every $t \geq 0$:

$$\Pr[\boldsymbol{X} \geq t\mathbf{E}[\boldsymbol{X}]] \leq \frac{1}{t}.$$

1. Markov's inequality is *false* for random variables that can take on negative values! Give an example.

   $X \sim \text{Unif}\{-1, +1\}$

   $\mathbb{E}[X] = 0$

   If $t = 10$

   $\mathbb{P}[X \geq 0] \leq \frac{1}{10}$ false should be $\mathbb{P}[X \geq 0] = \frac{1}{2}$

   $\frac{1}{2} \not\leq \frac{1}{10}$

2. Give a tight example for Markov's inequality. In particular, given μ and t, construct a random variable $\boldsymbol{X}$ such that $\mu = \mathbf{E}[\boldsymbol{X}]$ and $\Pr[\boldsymbol{X} \geq t\mu] = \frac{1}{t}$.

   $X = t\mu$ with probability $\frac{1}{t}$

   $X = 0$ with probability $1 - \frac{1}{t}$

   $\mathbb{P}[X \geq t\mu] = \frac{1}{t}$

Chebyshev's inequality. Let $\boldsymbol{X}$ be any random variable with well-defined variance[1], then

$$\Pr\left[|\boldsymbol{X} - \mathbf{E}[\boldsymbol{X}]| > t\sqrt{\mathbf{Var}[\boldsymbol{X}]}\right] \leq \frac{1}{t^2}.$$

To see the above inequality in action, consider the following problem:

> Let B be a bag with n balls, k of which are red and $n-k$ of which are blue. We do not have knowledge of k and wish to estimate k from ℓ independent samples (with replacement) drawn from B.

Let $\boldsymbol{X}$ be the number of red balls sampled.

1. What is $\mathbf{E}[\boldsymbol{X}]$?

X_i = indicator that i^{th} sample is red $\qquad X = \sum_{i=1}^{\ell} X_i$

$$\mathbb{E}[X] = \sum_{i=1}^{\ell} \mathbb{E}[X_i] = \sum_{i=1}^{\ell} \frac{k}{n} = \ell\frac{k}{n}$$

2. What is $\mathbf{Var}[\boldsymbol{X}]$?

Because trials are independent:

$$\mathrm{Var}(X) = \sum_{i=1}^{\ell} \mathrm{Var}(X_i) = \sum_{i=1}^{\ell} \mathbb{E}[X_i^2] - \mathbb{E}[X_i]^2 = \sum_{i=1}^{\ell} \frac{k}{n} - \left(\frac{k}{n}\right)^2$$

$$= \ell\left(\frac{k}{n}\right)\left(1 - \frac{k}{n}\right)$$

3. Choose a value for ℓ and give an algorithm that takes in n and $\boldsymbol{X}$ and outputs a number $\widetilde{k}$ such that $\widetilde{k} \in [k - \varepsilon\sqrt{k}, k + \varepsilon\sqrt{k}]$ with probability at least $1 - \delta$.

Estimate the number of red balls.

$$\mathbb{E}[X] = \ell\frac{k}{n}$$

Output $K = X\left(\frac{n}{\ell}\right)$ as your estimate. So that $\mathbb{E}[K] = \frac{n}{\ell}\mathbb{E}[X]$.

$$\mathrm{Var}(K) = \frac{n^2}{\ell^2}\mathrm{Var}[X] = \frac{n^2}{\ell^2}\ell\left(\frac{k}{n}\right)\left[1 - \frac{k}{n}\right) = \frac{kn}{\ell}\left(1 - \frac{k}{n}\right) \leq \frac{kn}{\ell}$$

$$\mathbb{P}[K \text{ deviates from range}] \leq \delta$$

$$\mathbb{P}\left[|K - \mathbb{E}[K]| > t\sqrt{\frac{kn}{\ell}}\right] \leq \frac{1}{t^2} \qquad \delta = \frac{1}{t^2}$$

$$t = \frac{1}{\sqrt{\delta}}$$

$$\mathbb{P}\left[|K - \mathbb{E}[K]| > \frac{1}{\sqrt{\delta}}\sqrt{\frac{kn}{\ell}}\right] \leq \delta$$

choose ℓ so that $\sqrt{\frac{n}{\delta\ell}} < \varepsilon$ $\qquad \varepsilon\sqrt{K}$

$$\ell > \frac{n}{\varepsilon^2\delta}$$

1 Estimating Votes

Suppose we have a stream of votes of form (Id, Yes) or (Id, No) which has person's Id (that is unique to them) and whether they voted Yes/No. We would like to estimate the fraction of Yes votes. Unfortunately, many people have voted multiple times. People who voted multiple times voted for the same option each time.

The Distinct Elements algorithm takes a stream as inputs and outputs $\tilde{n} \in [(1-\epsilon)n, (1+\epsilon)n]$ with probability $1-\delta$, where n the number of distinct elements seen in the stream, using small memory. Let $S(n, \epsilon, \delta)$ be the space complexity Distinct Elements uses in terms of n, ϵ, δ.

Using the Distinct Elements algorithm as a black box, provide an algorithm for estimating the fraction of "Yes" votes within a factor of, say, $(1+3\epsilon)$. You should count the vote given by each Id only once. State its space complexity in terms of S.

Challenge: Justify the error bound on your algorithm (you may assume $\epsilon < 1/3$ for simplicity).

Run Distinct Elements algo twice. Once for number of "yes" votes, once for total number of distinct votes.

Space complexity: $2S(n, \epsilon, \delta)$