CS 271: Randomness and Computation

Tyler Zhu

March 4, 2020

"A good stock of examples, as large as possible, is indispensable for a thorough understanding of any concept, and when I want to learn something new, I make it my first job to build one."

– Paul Halmos.

These are course notes for the Spring 2020 rendition of CS 271, Randomness and Computation, taught by Professor Alistair Sinclair.

Contents

1 Tuesday, January 21

Instructors for this course are Alistair Sinclair (sinclair@cs), OH in 677 Soda Mondays 1-2PM and Thursday 11-12AM, and Kush Bhatia (kush@) in 347 Alcove Soda Tuesdays 2-3PM and Wednesday 10-11AM.

Course website is at `https://people.eecs.berkeley.edu/~sinclair/cs271/s20.html`. There's no official textbook (but plenty of references); instead there is a set of fairly comprehensive lecture notes made available at the website.

We will assume familiarity with undergraduate algorithms, complexity theory, and data structures. CS 170 is approximately enough. There will be roughly four homeworks (each will be somewhat meaty), so don't expect to leave it to the last minute.

The philosophy of this class is to be more as a survey of techniques, rather than a deep dive into a specific topic. We'll look certain scenarios, and look at examples that use similar techniques. There will often be randomness (such as random graphs or random boolean circuits) because they sometimes serve as analogs for random inputs. Plus they're clean playgrounds for our purposes.

We can also view randomness in complexity theory in relation to resources (such as saving or minimizing the number of random bits).

1.1 Introduction

Typically we'll have an algorithm $\mathcal{A}$ which takes an input x and outputs an answer $\mathcal{A}(x)$. We can tweak this by also generating (as part of the algorithm) as input some sequence of random bits r, so our output is now $\mathcal{A}(x, r)$, i.e. dependent on both deterministic and random values. As a result, this output is also a random variable, so we can ask questions such as $\mathbb{P}[\mathcal{A}(x, r) = 0]$. Ideally this output isn't too far off from the true answer, otherwise this wouldn't be useful at all (just flip a coin).

We can interpret this randomness as being generated from a Turing machine, where at each of r timesteps we branch to choose either 0 or 1 as our bit, so there's 2^r possible outputs. While we can't exactly represent probabilities of say $\frac{1}{3}$ as a result, we can get arbitrarily close to it by approximation. In other words, all of our algorithms will have some errors associated with them.

1.2 Flavors of Randomized Algorithms

Here's a brief list of some of the randomized algorithms we'll see:

1. **Monte Carlo Algorithms**: Bounded error probability (output may be incorrect with small probability). These themsleves come in two flavors.
   (a) One-sided error: Yes answer is always right, but the No answer may sometimes be wrong. Formally,

$$\text{if true answer} = \text{“yes”} \implies \mathbb{P}[\mathcal{A} \text{ outputs “yes”}] \geq \epsilon > 0$$

$$\text{if true answer} = \text{“no”} \implies \mathbb{P}[\mathcal{A} \text{ outputs “no”}] = 1.$$

   This looks really bad, because in one case we have a guaranteed answer, but in the other case we have only an ϵ probability of being correct. In other words, probability of error (one-sided) is $\leq 1 - \epsilon$. But say we perform this test t times independently, and output "yes" if any trial outputs "yes", else output "no". Then

$$\mathbb{P}[\text{error}] \leq (1 - \epsilon)^t \leq \delta$$

if we take $t = O(\log 1/\delta)$ for your favorite choice of δ. The class of decision problems for which there exist one-sided error algorithms that run in polynomial time is RP, i.e. "randomized polynomial time." If the yes and no decisions are flipped, then the class becomes co-RP.

(b) Two-sided errors:

$$\text{if true answer} = \text{“yes”} \implies \mathbb{P}[\mathcal{A} \text{ outputs “yes”}] \geq \frac{1}{2} + \epsilon$$

$$\text{if true answer} = \text{“no”} \implies \mathbb{P}[\mathcal{A} \text{ outputs “yes”}] \leq \frac{1}{2} - \epsilon.$$

We need the epsilon here because we can do a similar trick from the above in order to concentrate the probability and make it stronger. Repeat t times and take a majority vote. One can use Chernoff/Hoeffding bounds to see this in general, but here we can do it from first principles.

In order to make a wrong decision, we need a majority of the outputs to be incorrect, so

$$\begin{aligned}
\mathbb{P}[\text{error in } t \text{ trials}] &= \mathbb{P}[\text{obtain } \leq \frac{t}{2} \text{ correct outputs}] \\
&\leq \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} (\frac{1}{2} + \epsilon)^i (\frac{1}{2} - \epsilon)^{t-i} \\
&\leq \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} (\frac{1}{2} + \epsilon)^{t/2} (\frac{1}{2} - \epsilon)^{t/2} \\
&= (\frac{1}{4} - \epsilon^2)^t \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} \leq (1 - 4\epsilon^2)^{t/2} \\
&= \delta
\end{aligned}$$

if we take $t = O(\log \delta^{-1})$. We refer to the class of decision problems that have a 2-sided error randomized algorithm in polynomial time as **BPP**.

2. **Las Vegas Algorithms**: Output is always correct, and the *expected* running time is bounded. Equivalently, the running time is bounded and the output is always correct, but the algorithm may output "?" with probability $\leq 1/2$.

Exercise 1.1. Check that these are equivalent formulations.

Proof. Say the expected running time is t. Then run it for $2t$ steps, and by Markov's inequality, the probability it hasn't completed is $1/2$. Conversely, if we have the latter scenario, then with probability 1 it completes running. Say it has probability $1/2$ of outputting an answer by time t: then the overall time is $\frac{1}{2}t + \frac{1}{4}(2t) + \dots$ which coverges to t. □

The class of decision problems with a Las Vegas algorithm with polynomial bounded expected running time is **ZPP**, i.e. zero-error polynomial time.

We can draw a picture of these complexity classes. We would draw RP and co-RP, and ZPP would lie somewhere in the middle. In fact, ZPP is *exactly* the intersection between these two classes.

Exercise 1.2. Prove that ZPP = RP ∩ co-RP.

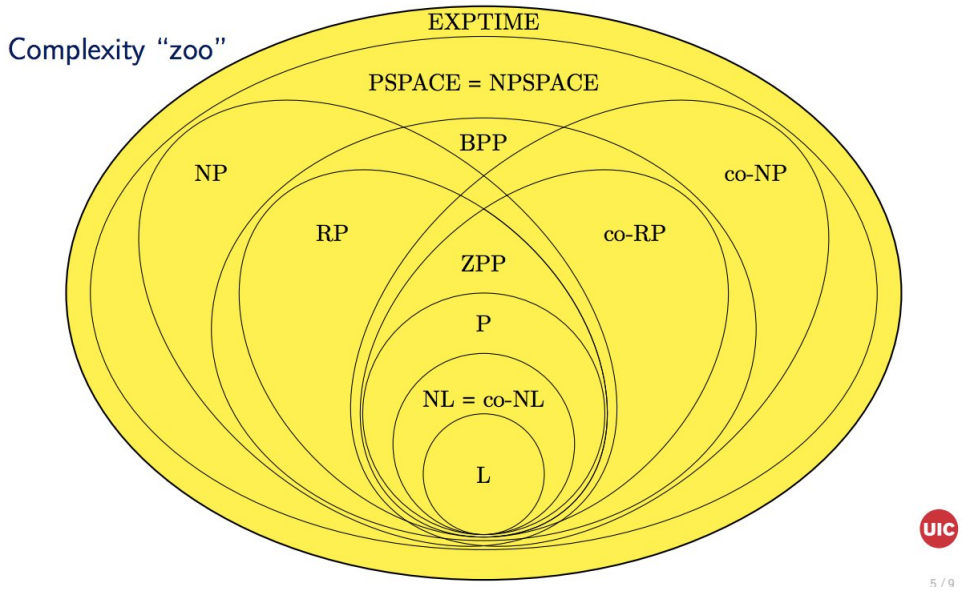

Figure 1: The big picture.

BPP contains everything (and more), while P would be inside ZPP and NP would contain RP and some of BPP.

The open questions in this field are if P = RP and if P = BPP; in otherwords, we want to know if randomness buys you anything.

- If P = BPP, then we don't gain much from randomness, but it would not imply P = NP.
- If P = NP, then everything collapses (including BPP). This follows from a result in compelexity classes, where it's known that BPP $\subseteq \text{NP}^{\text{NP}}$ (the second order of NP). In this case, NP^{NP} = P.
- If $NP \subseteq BPP$, then NP has polynomial size classes (very unlikely).

Here's a remarkable theorem which, out of two things we'd both like to be true, states that one of them must be false.

Theorem 1 (Impagliazzo-Wigderson)**.** If SAT cannot be solved by circuits of size $2^{o(n)}$, then BPP = P.

In other words, if what we believe about NP algorithms is true (i.e. we should need exponentially amount of work to solve SAT), then randomness doesn't actually buy you anything. But if it turns out randomness does help, then there exist sub-exponential sized circuits to solve SAT. Both are remarkable statements.

This also doesn't imply that P = NP quite yet, as there's some tricky point about how your circuits grow with your inputs, but that's a problem for another class.

Here's another philosophical question. Suppose we do show that randomness does help. We still have the following problem: say you're implementing a randomized algorithm. Where do you get your random bits? Random number generators are usually pseudo-random, i.e. are pregenerated, but we also can't tell the difference between truly random algorithms and pseudo-random ones. If BPP is different from P, then we wouldn't be able to simulate algorithms that express this difference. We would need a source of true randomness.

Most complexity theorists bet that BPP = P, and even though conceptually these algorithms are most clean, there are most likely going to be huge overheads that would make them infeasible to use realistically.

1.3 Examples of Randomized Algorithms: Matrix Multiplication, Associativity

In the following examples, we want to be able to check if the answer is correct without having to do the same amount of work as the computer did. Philosophically, checking should be easier than doing. With the help of randomness, we can do this.

Example 1.3 (Check Matrix Multiplication)**.** Suppose we have as input three $n \times n$ matrices A, B, C.

Question. Is $AB = C$?

Algorithm: Pick a random vector $(r_1, r_2, \ldots, r_n)$ such that each r_i is chosen independentaly and uniformly at random from a set S, $|S| \geq 2$. If $A(Br) \neq Cr$, then output "no", else output yes. This is faster, i.e. $O(n^2)$, as we're only doing matrix-vector multiplication (unless matrix multiplication achieves $O(n^2)$ time).

This is an example of what we call a *Witness search*, as we're searching for a vector which is a witness that $AB \neq C$. Clearly this is a one-bounded algorithm.

Claim: If $AB \neq C$, then $\mathbb{P}[(AB)r = Cr] \leq \frac{1}{|S|}$.

Proof. Define $D = AB - C$. Suppose $D \neq 0$, and WLOG $d_{11} \neq 0$, i.e. the upper-left corner is not 0.

Let's look at the first vector of D. We know that

$$(Dr)_1 = \sum_{i=1}^{n} d_{1i} r_i = 0$$

$$r_1 = -\frac{1}{d_{11}} \sum_{i=2}^{n} d_{1i} r_i$$

So the probability that r_1 is the one number that makes this entire sum 0 is $\leq \frac{1}{|S|} \leq \frac{1}{2}$. □

Example 1.4 (Testing Associativity (Rajagoparlan/Schulman))**.** Suppose we have as input a binary operator $\circ$ on a set X of size n.

Question. Is $\circ$ associative, i.e. $(i \circ j) \circ k = i \circ (j \circ k)$ for $i, j, k \in X$?

One naive idea is to just test triples for associativity, but it turns out there exist binary operators which are mostly associative except for sometimes a *constant* amount of witnesses. Instead, the key idea will be to test subsets.

Algorithm: Consider *sets* of elements $R, S, T \in 2^X$. Pick R, S, T independently and u.a.r. over 2^X. If $(R \circ S) \circ T \neq R \circ (S \circ T)$, then output "no", else output "yes".

This works because of the following fact:

Exercise 1.5. Check that $\circ$ is associative over X iff $R \circ (S \circ T) = (R \circ S) \circ T \ \ \forall R, S, T \in 2^X$.

We can do this check in $O(n^2)$ (for all members of the subset) using vector dot products (think of elements as tuple (r_i, i) where r_i is 1 if $i \in R$, and 0 else). In this case, we can create a giant lookup table of products and reference from there, so computing products is constant time and creating this table is $O(n^2)$. This is faster than the $O(n^3)$ needed to check all triples of X.

The heart of the algorithm is the following claim, which is a clever application of the principle of inclusion-exclusion.

Claim. If $\circ$ is not associative, then at least 1/8 of the tuples (R, S, T) are witnesses.

Proof. Partition 2^X into disjoint groups of 8. Assume $\circ$ is not associative, so there exists $i^*, j^8, k^* \in X$ such that $(i^* \circ j^*) \circ k^* = i^* \circ (j^* \circ k^*)$.

Let R_0, S_0, T_0 be any subsets s.t. $i^*, j^*, k^* \notin R_0, S_0, T_0$. Let $R_1 = R_0 \cup \{i^*\}$, $S_1 = S_0 \cup \{j^*\}$, $T_1 = T_0 \cup \{k^*\}$. Now we have 8 triples $\{(R_a, S_a, T_a) | a \in \{0, 1\}\}$. Define $f(R, S, T) = |(R \circ S) \circ$

$T - R \circ (S \circ T)|$, i.e. the number of witnesses in each triplet of sets. Then by inclusion-exclusion, we can write

$$\begin{aligned} f(i^*, j^*, k^*) = f(R_1, S_1, T_1) \\ - f(R_0, S_1, T_1) - f(R_1, S_0, T_1) - f(R_1, S_1, T_0) \\ + f(R_0, S_0, T_1) + f(R_0, S_1, T_0) + f(R_1, S_0, T_0) \\ - f(R_0, S_0, T_0). \end{aligned}$$

Since we know that the LHS is nonzero, at least one of the terms on the right has to be nonzero as well, from which our claim follows. □

2 Thursday, January 23rd

2.1 More Randomized Algorithms: PIT, Perfect Matchings

Example 2.1 (Polynomial Identity Testing)**.** Suppose we have as input a polynomial $Q(x_1, \ldots, x_n)$.

Question. Is $Q = 0$?

To understand the motivation for this, suppose we wanted to ask if $(x_1+x_2)(x_1-x_2) = x_1^2-x_2^2$. Then we could make $Q(x_1, x_2)$ the difference of the two polynomials to test for equality.

For a more complicated example, consider the Vandermonde determinant

$$\det \begin{pmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_n \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1} \end{pmatrix} = \prod_{i<j}(x_j - x_i).$$

It's very nontrivial to see if these two polynomials are the same.

Algorithm (Schwartz-Zippel): Pick a random point $(r_1, \ldots, r_n)$ where each r_i is chosen u.a.r. from a set S. Test if $Q(r_1, \ldots, r_n) = 0$.

Claim. If $Q(x_1, \ldots, x_n)$ is a non-zero polynomial of degree d, then $\mathbb{P}[Q(r_1, \ldots, r_n) = 0] \leq d/|S|$.

Proof. Induction on the number of variables n.

Base case: $n = 1$, which is the univariate case and follows from the Fundamental Theorem of Algebra.

Inductive step: Write

$$Q(x_1, \ldots, x_n) = A(x_2, \ldots, x_n)x_1^k + B(x_1, \ldots, x_n).$$

Assume k is the largest power of x_1, so that $\deg(A) \leq d - k$ and degree of x_1 in B is $< k$.

Assume $x_2 = r_2, \ldots, x_n = r_n$ are chosen first. Let $\mathcal{E}$ be the event that $A(r_2, \ldots, r_n) = 0$. We have two cases:

(a) $\mathcal{E}$ happens, which by induction has probability $\mathbb{P}[\mathcal{E}] \leq (d - k.)/|S|$.

(b) $\mathcal{E}$ doesn't happen. So $A(\cdot)x_1^k + B(\cdot)$ is a non-zero univariate polynomial in x_1. Also, since $\deg B < k$ (as we pulled out the highest powers into A), $\mathbb{P}[Q(r_1, \cdots, r_n) = 0|\neg\mathcal{E}] \leq k/|S|$ by base case.

Summing them up, we have that

$$\mathbb{P}[Q = 0] = \mathbb{P}[Q = 0|\mathcal{E}]\mathbb{P}[\mathcal{E}] + \mathbb{P}[Q = 0|\neg\mathcal{E}]\mathbb{P}[\neg\mathcal{E}] \leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}.$$

□

This is a remarkable algorithm because it convinced people that randomness is useful. Without randomness, we have no clue how to approach this problem.

Question. Can this algorithm be derandomized?

Theorem 2 (Impagliazzo-?)**.** If PIT can be solved determinstically in poly time then either

(1) The permanent[1] cannot be computed by poly size circuits, OR

(2) NEXP does not have poly size circuits.

[1]The *permanent* is the determinant but without the $\text{sgn}(\sigma)$ factor in the expansion; see later in this section.

Recall that a *matching* in a bipartite graph $\mathcal{G} = (V_1, V_2, E)$ is a collection of edges so that no vertex is in more than one edge in that collection. A *perfect matching* is a matching which contains every single vertex.

Recall that there are algorithms to determine if a graph has a perfect matching (and to find one), such as Edmonds-Karp flow based algorithm. But none are very efficient, so instead we'll develop a randomized one.

Example 2.2 (Perfect Matchings)**.** Suppose we have a bipartite graph $\mathcal{G} = (V_1, V_2, E)$ as input.

Question. Does there exist a perfect matching in $\mathcal{G}$?

Given $\mathcal{G}$, construct the *Tutte matrix* $A_\mathcal{G}$ which is an $|V_1| \times |V_2|$ matrix where the indeterminate x_{ij} is in the ij-entry if there is an edge from $i \in V_1$ to $j \in V_2$ and zero otherwise.

Claim. $\mathcal{G}$ has a perfect matching iff $\det(A_\mathcal{G}) \neq 0$.

Proof. We expand the determinant as

$$\det(A_\mathcal{G}) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}.$$

Now apply the Schwartz-Zippel algorithm to test if $\det(A_\mathcal{G}) \neq 0$ as a polynomial. The degree is n so we need to plug in random values distributed uniformly in $\{1, 2, \ldots, 2n\}$. □

We can also extend this result to general graphs (not just bipartite)

Example 2.3 (Perfect Matchings for general graphs)**.** Instead of using the Tutte matrix, we'd use the skew-symmetrix matrix

$$B_\mathcal{G} = \begin{pmatrix} 0 & & & \\ & \ddots & -x_{ij} & \\ & x_{ij} & \ddots & \\ & & & 0 \end{pmatrix}.$$

We can make the same claim here, which is that $\det(B_\mathcal{G}) \neq 0$ iff $\mathcal{G}$ has a perfect matching.

2.2 Parallelizing Algorithms

We can efficiently parallelize such algorithms. There's a complexity class called NC where we can solve problems in polylog(n) time and poly(n) processors, and it's a fact that computing determinants is in NC. As an aside, NC⊆P.

There's a naive idea for parallelizing this decision problem to determine perfect matchings. Remove an edge, run the algorithm, and recursively run until we determine edges that must be in the perfect matchings and repeat.

This is parallelizable across choices of edges, but the subprocesses can't be run in parallel. One pitfall however is that not only are we doing work in cases multiple times, but that we may have *multiple* perfect matchings, so our processes may be working on different matchings on the same time. If we want only one matching, then this is a waste of effort!

There's a lemma to help us with this however, due to Mulmanley/Vazirani/Vazirani (both the professor at Berkeley *and* his brother).

Lemma 3 (Isolation Lemma)**.** Let $S_1, \ldots, S_k$ be arbitrary subsets of a set S of cardinality m. Let each element $x \in S$ have a weight w_x chosen independently and u.a.r. from $\{1, \ldots, l\}$. Then

$$\mathbb{P}[\exists \text{ unique min. weight set } S_i] \geq \left(1 - \frac{1}{l}\right)^m \geq 1 - \frac{m}{l}.$$

This is enough for our purposes of randomization as long as we take $l = 2m$ approximately so that the probability is greater than $1/2$. The proof is due to a high school student in Israel. [2]

Proof due to Ta-Shma. Assume w.l.o.g. that $S_i \not\subseteq S_j$ for any i, j. Let $\mathcal{W}$ denote the set of all weight functions $\{w_x\}$, and $\mathcal{W}^+$ denote the set of all weight functions such that $w_x > 1$ for all $x \in S$. The proof rests on the following claim.

Claim. For each function $w \in \mathcal{W}^+$, we can associate a *distinct* function $w' \in \mathcal{W}$ such that w' has a unique minimum weight set. This implies

$$\mathbb{P}[\exists \text{ unique minimum weight set}] \geq \frac{|\mathcal{W}^+|}{|\mathcal{W}|} = \left(1 - \frac{1}{l}\right)^m$$

as we claimed above.

We prove the claim as follows. Take a function $w \in \mathcal{W}$ - pick *any* minimum weight set S_i under w. Construct w' via

$$w'_x = \begin{cases} w_x - 1 & \text{if } x \in S_i \\ w_x & \text{otherwise} \end{cases}.$$

Then w' has unique minimum weight set and can recover w from w'. □

Using this, we can create an algorithm for finding a perfect matching in a bipartite graph in parallel.

(a) Set each edge weight w_{ij} as in the Isolation Lemma (with $S = E, l \geq 2|E|$).

(b) By the Isolation Lemma, $\mathcal{G}$ has min. weight. perfect matching with probability $\geq 1/2$.

(c) Plug into Tutte matrix the values $2^{w_{ij}}$ (rather than x_{ij}) - call this matrix A.

- Calculate 2^w, the largest power of 2 that divides $\det(A)$, which really looks like $\sum_M (\pm 1) 2^{w(M)}$.
- For each edge (i, j) in parallel:
  - compute $t_{ij} = 2^{w_{ij}} \det(A_{ij})$ where $A_{ij} = \{i, j\} -$ minor of A.
  - include (i, j) in perfect matching iff largest power of 2 that divides t_{ij} is 2^w.
- Check that the selected edges from a perfect matching and output it if so.

There's a "quasi-parallel" algorithm due to Svensson and Tarnaski for perfect matchings in polylog time and $n^{\text{poly}\log(n)}$ processors by derandomizing the Isolation Lemma by exploiting structure in perfect matchings. Anan Nazirani also has an NC algorithm for planar graphs.

[2] What were you doing when you were in high school?

3 Tuesday, January 28th

We'll talk a lot about primes today, including applications to testing equality of numbers and primality testing.

3.1 Fingerprinting

Example 3.1 (Testing Equality of Large Numbers)**.** Alice and Bob are on different planets, and they each of a large piece of data. They want to test if their data is the same. Let their data be n-bit numbers a and b respectively where n is *very* large. We want to test if $a = b$.

Of course Alice could send a to Bob, but this is probably prohibitively expensive. What we'll show instead is that Alice will send a fingerprint to Bob, and Bob will determine (w.h.p.) is b is equal to a based on that.

Algorithm: Alice picks a random prime $p \in \{1, 2, \ldots, T\}$ (for a small T) and computes a fingerprint $F_p(a) = a \mod p$ and sends both $F_p(a)$ and p. Bob then computes his fingerprint, and if $F_p(a) = F_p(b)$ he responds Yes, otherwise he responds No.

Note that this is a one-sided error: if $a = b$, then Bob will always respond Yes, and otherwise he responds No with some probability based on the T we pick. We will analyze this latter case.

Suppose $a \neq b$ and $F_p(a) = F_p(b)$. Then $p|(a - b)$ (where $a - b$ is an n-bit number). Crudely, the number of primes dividing $a - b$ is at most n (as $\log(a - b)$ is a good lower bound). Now we make use of the following classical fact:

Theorem 4 (Prime Number Theorem)**.** Define $\pi(x) = \#$ of primes $\leq x$ to be the prime counting function. Then,

$$\frac{x}{\ln x} \leq \pi(x) \leq 1.26\frac{x}{\ln x} \quad \forall\, x \geq 17.$$

Thus, we can compute

$$[\#\text{ of primes} \leq T] \approx \frac{T}{\ln T}$$

and thus

$$\mathbb{P}[\text{error}] \leq \frac{n \ln T}{T}.$$

So we should take $T = cn$ so that $\mathbb{P}[\text{error}] = \frac{1}{c} + o(1)$.

In fact, we can make a much more refined upper bound by using $\pi(x)$ an an upper bound on the number of primes that divide $a - b$. This gives

$$\mathbb{P}[\text{error}] \leq \pi(n)\frac{\ln T}{T} \leq 1.26\frac{n}{\ln n}\frac{\ln T}{T}.$$

The hard part of this example then is to pick a random prime p. The PNT guarantees that the density of primes is quite decent in this range, and we'll look at algorithms for testing primality later in today's lecture, so Alice can just keep trying random numbers.

Example 3.2 (Pattern Matching)**.** Suppose we have a source text $X = x_1x_2 \ldots x_n$ and a pattern $Y = y_1y_2 \ldots y_m$, $m < n$. We want to determine if Y occurs in X.

There's an obvious $O(mn)$ brute force algorithm, but there's also some classical results in algorithms such as Boyer/Monse or Knuth/Morris/Pratt that solve this in $O(m + n)$. There is also a nice randomized algorithm in $O(m + n)$ found by Rabin/Karp. The main idea is to maintain a sliding window and compare finger prints of X to the fingerprint of Y.

Algorithm: Pick a random prime $p \in \{1, 2, \ldots, T\}$. Compute $F_p(Y) = Y \mod p$. Then, for $j = 1$ to $n - m + 1$:

- compute $F_p(X_j)$
- if $F_p(X_j) = F_p(Y)$ output "match" and halt.

otherwise output "no match."

How big should T be? We just use our previously calculated bound so that

$$\mathbb{P}[\text{error}] \leq n\frac{\pi(m)}{\pi(T)}.$$

We can do better though. For p to be *bad*, it must divide one of the $Y - X_j$'s, so it also divides

$$\prod_{j=1}^{n-m+1} (Y - X_j) \leq nm \text{ bits },$$

so it suffices to take $T = cnm$ which gives a $O(\log n)$ bit fingerprint.

One issue is how we would compute each of the X_j's efficiently. Using our sliding window trick, we can compute each next X_{j+1} based on the previous one efficiently since

$$X_{j+1} = 2(X_j - 2^{n-1}x_j) + x_{j+m}$$

so

$$F_p(X_{j+1}) = 2(F_p(X_j) - 2^{n-1}x_j) + x_{j+m}.$$

Updating our fingerprints only takes $O(1)$ time then.

3.2 Primality Testing

Example 3.3 (Primality Testing)**.** We have an integer n as input (potentially hundreds or thousands of bits), and we want to determine if n is prime.

You might try a prime number sieve, but those are typically polynomial in n, which is infeasible for our purposes. Another idea is to test divisors, but large numbers typically have very few divisors, so this will be inefficient.

One good idea is to start with Fermat's Little Theorem, which is the following:

Theorem 5. If p is prime then $a^{p-1} \equiv 1$ for all $a \in \{1, 2, \ldots, p-1\}$.

Using this, we can pick a random number a and see if $a^{p-1} \equiv 1$. If this is true, then we output prime, and otherwise we output not prime. This is called the *Fermat test* and is actually used in practice.

Unfortunately, there are numbers for which $a^{p-1} \equiv 1$ yet are not primes; these are known as the Carmichael numbers, and we'll talk about them later.

A better insight is the following:

Fact 6. Suppose n is not prime and not *all* a's satisfy $a^{n-1} \equiv 1 \mod n$. Then $\mathbb{P}[\text{error}] \leq 1/2$.

This is because the group of witnesses is a subgroup of the multiplicative group $Z_n^\times$ on n number (multiplicative group of integers $\mod n$ s.t. $\gcd(a, n) = 1$), so we can apply Lagrange's Theorem.

Let $S_n = \{a \in \mathbb{Z}^\times \text{ s.t. } a^{n-1} \equiv 1 \mod n\}$. Then this is a proper subgroup if $S_n \ncong \mathbb{Z}^\times$ (which happens if n is a Carmichael number), and its order cannot be more than half of that of $\mathbb{Z}^\times$. This is also the set of all witnesses, so our result follows.

To do better, we need a better witness test to weed out Carmichael numbers.

Theorem 7 (Miller/Rabin Primality Test)**.** Witnesses to compositeness of n are non-trivial square roots p. Suppose p is prime. Then there exists exactly two solutions to $x^2 - 1$ in $\mathbb{Z}_p$. If we can find a number a s.t. $a^2 \equiv 1$, $a \neq \pm 1$ in $\mathbb{Z}^\times$ then n is not prime.

For example, if $n = 7, 8$, then try $a = 7$.

If $n = 2$ is even or a perfect power then output composite. Otherwise, compute r, p such that $n - 1 = 2^r p$ (where p is odd).

Then pick $a \in \{1, \ldots, n-1\}$ u.a.r. If $\gcd(a, n) \neq 1$, then output composite. Otherwise, compute $b_i = a^{2^i p}$ for $i = 0, 1, \ldots, r$ so that this sequence is

$$a^p, a^{2p}, \ldots, a^{2^r p} = a^{n-1}.$$

If $b_r \neq 1$ then output composite. Otherwise, if $b_0 = 1$ then output "prime?". Otherwise, let $j = \max\{i : b_i \neq 1\}$. If $b_j \neq -1$ then output composite, otherwise output "prime?". One should try this when $n = 561$ and $a = 2$.

Claim. If n is composite, odd, and not a prime power, then $\mathbb{P}[a \text{ is not a witness}] \leq 1/2$.

Proof. Show that non-witnesses lie in a proper subgroup by using Lagrange's theorem.

Say that s is a *bad power* if $x^s \equiv \pm 1 \mod n$ for some $x \in \mathbb{Z}_n^\times$. We claim that

$$H_s = \{x \in \mathbb{Z}_n^\times | x^s = \pm 1 \mod n\}$$

is a proper subgroup of $\mathbb{Z}_n^\times$ for any bad power s.

Before we prove this claim, let's see how we'd finish with it. Apply this claim to H_{s^*} where s^* is the largest bad power. Suppose a is a non-witness. Then either

$$a^r = a^{2r} = \cdots = a^{n-1} = 1 \mod n$$

or

$$a^{2r} = \cdots = -1 \mod n$$

in both cases, $a \in H_{s^*}$.

Now we prove the claim. We will show it explicitly, that there exists $y \in \mathbb{Z}_n^\times \setminus H_S$. Since s is a bad power, $\exists x \in \mathbb{Z}_n^\times$ such that $x^s \equiv -1 \mod n$. Since n is odd, composite, and not a power, we can find n_1, n_2 odd such that $n = n_1 n_2$ and $\gcd(n_1, n_2) = 1$.

By the CRT, there exists a unique $y \in \mathbb{Z}_n$ such that

$$\begin{aligned} y &= x \pmod{n_1} \\ y &= 1 \pmod{n_2} \end{aligned}$$

We claim that $y \in \mathbb{Z}_n^\times \setminus H_s$. Since $y = x \mod n$, $\gcd(x, n) = 1$ so $\gcd(y, n_1) = \gcd(x, n_1) = 1$. Also $\gcd(y, n_2) = 1$, so $\gcd(y, n) = 1$ so $y \in \mathbb{Z}_n^\times$.

Furthermore, $y^s \equiv x^s \equiv -1 \mod n_1$ and $y^s \equiv 1 \mod n_2$, but neither implies that $y^s \equiv \pm 1 \mod n$ (by CRT), so $y \notin H_s$. □

4 Thursday, January 30th

Last time we saw an algorithm for primality testing which runs in poly($\log n$) time. One of the great hallmarks of randomized algorithms theory was figuring out how to derandomize another primality testing algorithm (a result in 2002). There is a randomized algorithm proposed by Agrawal/Biswas in '99 which was derandomized by Agrawal/Kayal/Saxena in '02. The theorem is mostly number theory however, so we'll skip it for now. Instead, we'll look at their randomized algorithm.

Miller found that under the extended Riemann Hypothesis, we can derandomize the Miller-Rabin algorithm by deterministically finding a choices of a: in fact, within the first $O(\log^2 n)$ choices for a, there exists a witness.

4.1 Primality Testing Polynomial

We discuss the algorithm proposed by Agrawal/Biswas in '99 for primality testing, which makes use of polynomials. It makes use of the following fact.

Fact 8. For all $a > 1$ such that $\gcd(a, n) = 1$, $n > 2$ is prime iff $(x - a)^n \equiv x^n - a \mod n$.

Proof. Binomial Theorem. All of the intermediate binomial coefficients go to zero if n is prime, and otherwise stick around. □

Knowing this fact, one idea might be to use Schwartz-Zippel to test if these polynomials are equivalent. But there's two problems with that: the first is that we need to work over a field, which we might not if n isn't prime, and the second is that we need the number of points we select to be much greater than the degree of the polynomial, which it is not as our degree is the size of our ring.

Instead of searching on a, we're going to do something smarter.

Algorithm: If n has a divisor < 17, or if n is a perfect power, then output "composite."

Let $d = \lceil \log n \rceil$, and let $r(x) = x^d + r_{d-1}x^{d-1} + \cdots + r_1 x + r_0$ be a random polynomial where the coefficients r_i are chosen independently and u.a.r. from $\mathbb{Z}_n$. If $(x + 1)^n \not\equiv x^n + 1 \mod (r(x), n)$ then output "composite", otherwise output "prime?".

Claim. If n is composite, has no divisor < 17 and is not a prime power, then

$$\mathbb{P}[(x + 1)^n \not\equiv x^n + 1 \mod (r(x), n)] \geq \Omega(1/\log n).$$

Proof. We want to work mod a prime number, but n isn't necessarily prime. By assumption, n has a prime divisor $p \geq 17$. Define $Q(x) = (x + 1)^n - (x^n + 1)$. If $Q(x) \not\equiv 0 \mod p$ then $Q(x) \not\equiv 0 \mod n$, so we can search for witnesses over $\mathbb{F}_p$ instead of mod n (this is a nontrivial fact, but similar to our earlier one).

A polynomial $r(x)$ is a *witness* if $r(x)$ is *not* an irreducible factor of $Q(x) \mod p$ (which means $Q(x)$ has non trivial factors, and hence cannot be 0). So

$$\begin{aligned}\mathbb{P}[r(x) \text{ is a witness }] &\geq \mathbb{P}[r(x) \text{ is irreducible and not a factor of } Q(x)] \\ &= \mathbb{P}[r(x) \text{ is irreducible}] - \mathbb{P}[r(x) \text{ is an irred factor of } Q(x)]\end{aligned}$$

Breaking it down,

$$\mathbb{P}[r(x) \text{ is irreducible}] \geq \frac{\#\text{ of monic irred. polys of deg } d \text{ over } \mathbb{Z}_p}{\text{total no. of monic polys of deg } d} \geq \frac{p^d - \sqrt{p}}{p^d} \geq \frac{1}{2d}$$

since $p \geq 17$, thanks to some nontrivial results in polynomial theory. Also,

$$\mathbb{P}[r(x) \text{ is irreducible factor of } Q(x)] \leq \frac{n/d}{p^d} \leq \frac{1}{4d}$$

using $d = \log n$ and $p \geq 17$.

Thus,

$$\mathbb{P}[r(x) \text{ is a witness}] \geq \frac{1}{2d} - \frac{1}{4d} = \frac{1}{4d} = \Omega(1/\log n).$$

□

AKS derandomized this by instead of fixing a and searching over witnesses $r(x)$, they fix $r(x) = x^d - 1$ and search over a to find $(x-a)^n \not\equiv x^n - a \mod (r(x), n)$. Then one can show that if n is composite, there exists a witness within the first $O(\log^2 n)$.

4.2 The Probabilistic Method

This tool was invented by the Hungarian mathematician Paul Erdos, one of the most prolific mathematicians of all time. Initially it didn't start out as an algorithmic tool, but eventually we'll see algorithms that it creates.

The key idea here is instead of considering a specific structure, we pick a random element of a family of structures and show that the probability it has a desired property is greater than 0. This means that there must exist some such structure (but is not constructive). If we can show that this probability is $> 1/2$, then within any collection, there is a high probability that one of the structures has our desired property.

We'll look at the first application that this method was invented for: Ramsey Theory [Erdos 1947]

Definition 9. The kth (diagonal) Ramsey number R_k is the smallest n such that in any 2-coloring of K_n, we must have a monochromatic k-clique.

For example, when $n = 5$ and $k = 3$, we can color the outer edges of K_5 with red and everything else with blue, so there is no triangle persent.

Can we do the same with $n = 6$ and $k = 3$? We claim no. Every vertex has degree 5, so at least three edges must be the same color, say red. Then the edges between the vertices they point to must be the opposite color to avoid making a triangle, which is blue. But these blue edges form a triangle!

There are also asymmetric Ramsey numbers, where $R(k, l)$ is the smallest n so that we must have a k-clique or an l-independent set, i.e. either k people that all know each other or l people none of whom know each other.

These Ramsey numbers are quite difficult to compute; we only know $R_3 = 6$ and $R_4 = 18$, and R_5 is something we have yet to figure out. The only bounds we have are

$$2^{k/2} \leq R_k \leq 2^{2k-3} \approx 4^k.$$

The upper bound is obtained with induction, and there are still no better bounds, but the lower bound is obtain by the probabilistic method, which we'll show now.

Theorem 10 (Ramsey Number Bound)**.** $R_k \geq 2^{k/2}$.

Proof. Color the edges of K_n independently and u.a.r. red or blue. Let C be any k-clique in K_n. Then

$$\mathbb{P}[C \text{ is monochromatic}] = 2 \cdot 2^{-\binom{k}{2}} = 2^{1-\binom{k}{2}}.$$

Thus, by the union bound,

$$\begin{aligned}\mathbb{P}[K_n \text{ has a monochr. clique}] &\leq \binom{n}{k} 2^{1-\binom{k}{2}} \\ &\leq \frac{n^k}{k!} 2^{1-\binom{k}{2}}\end{aligned}$$

So choose n such that this is < 1. Set $n = 2^{k/2}$. Then this probability is upper bounded by

$$\frac{1}{k!} 2^{\frac{k+2}{2}} < 1$$

for $k \geq 3$. So there exists a coloring such that k_n has no monochromatic clique. □

Here's another surprising application of the Probabilistic Method. The min-cut problem is a problem we can solve efficiently using flow techniques while the max-cut problem, finding the cut of maximize size in a graph $G = (V, E)$, is NP-hard.

The probabilistic method gives us a ten-second proof of a surprising lower bound to the maximum cut size.

Theorem 11 (Max-Cut Bound)**.** Any graph G contains a cut of size $\geq \frac{|E|}{2}$.

Proof. Pick a random cut by placing each vertex on either set L or R with probability $\frac{1}{2}$. Thus for each edge e, $\mathbb{P}[e \text{ is in cut}] = 1/2$.

Let $X =$ the total number of cut edges. Then

$$X = \sum_{e \in E} X_e \quad \text{where } X_e = \begin{cases} 1 & \text{if } e \text{ in cut} \\ 0 & \text{else} \end{cases}$$

and

$$\mathbb{E}[X] = \sum_{e \in E} \mathbb{E}[X_e] = \frac{1}{2}|E|.$$

Hence there is a cut such that $X \geq \frac{1}{2}|E|$. □

There's another NP-hard problem of finding the maximum independent set of vertices in $G = (V, E)$ (set of vertices with no edges between them).

Theorem 12 (Independent Set Bound)**.** Every graph has an independent set of size $\geq \sum_{v \in V} \frac{1}{\deg(i)+1}$.

For example, every 3-regular graph has an independent set of size $\geq |V|/4$.

Proof. Assign random value $\alpha_v \in [0, 1]$ to each vertex v u.a.r. Then the set of "local minima" is an independent set. But

$$\mathbb{P}[v \text{ is a local min}] = \frac{1}{\deg(v) + 1}$$

since there are $\deg(v) + 1$ connected to v, so by linearity of expectation,

$$\mathbb{E}[\text{size of indep. set}] = \sum_{v \in V} \frac{1}{\deg(v) + 1}.$$

□

One last example is a proof from the book, and one of Sinclair's favorite.

Definition 13. The *Crossing number* of a graph $G(V, E)$ is the minimum number of edge crossings in any drawing of G in the plane.

Obviously every planar graph has crossing number 0. Euler's formula, i.e. that $3n - 6 \geq e$ when $c = 0$, gives this fact.

Fact 14. $c \geq m - 3n + 6$.

Using the probabilistic method, we can get a much more powerful result due to Chazelle/Sharir/Welzt.

Theorem 15 (Graph Crossing Number)**.** For all $m \geq 4n$, $c \geq \frac{m^3}{64n^2}$.

For values of m very close to $3n$, i.e. almost planar, Euler's formula gives a better bound. But this theorem gives us a much better bound in most other cases.

Proof. Fix an optimal planar drawing of G with c crossings. Construct a random induced subgraph G' by including each vertex of G with probability p independently - retain only edges between included vertices.

Let n_p, m_p, c_p be the number of vertices, edges and crossings that remain. Apply the Euler bound to the remaining graph G', so

$$c_p \geq m_p - 3n_p + 6 \geq m_p - 3n_p.$$

Take expectations, so that

$$\mathbb{E}[c_p] \geq \mathbb{E}[m_p] - 3\mathbb{E}[n_p].$$

But these expectations are easy: $\mathbb{E}[n_p] = np$, $\mathbb{E}[m_p] = mp^2$, and $\mathbb{E}[c_p] = cp^4$ since we need one, two and four vertices to remain in G'. Then

$$cp^4 \geq mp^2 - 3np \implies c \geq \frac{m}{p^2} - \frac{3n}{p^3}.$$

Now we just need to do a bit of calculus to pick the p that maximizes this bound. It turns out that we choose $p = 4n/m$ to maximise, which gives

$$c \geq \frac{m^3}{64n^2}.$$

Of course, we need to make sure $p < 1$, which is why we assume that $m \geq 4n$. □

5 Tuesday, February 4th

5.1 Unbalancing Lights

Goal: Given an arbitrary initial configuration of an $n \times n$ grid of lights, set the switches with row and column flips so as to maximize the # of lights on. Missed this proof, but a lot of probability and a lot of methods.

5.2 Graphs with Large Girth and Large Chromatic Number

Recall that the *girth* of a graph G is the size of the smallest cycle in G. Intuitively, a larger girth means it would be easier to color graphs since the graph is more sparse and thus would need less colors. Erdos's classic result shows this is not the case.

Theorem 16. For any integers k, l, there exists a graph with girth $\geq l$ and chromatic number $\geq k$.

Proof. Construct a random graph $G \in \mathcal{G}(n,p)$ with $p = n^{1/l-1}$. Then

$$\mathbb{E}[G_E] = \binom{n}{2} p \approx \frac{1}{2} n^{1+1/l}.$$

So these are very sparse graphs, but the edges still grow slightly faster than linear.

Let a random variable $X =$ the number of cycles of length $< l$ in G. Then we can count all of the such cycles by

$$\begin{aligned}\mathbb{E}[X] &= \sum_{i=3}^{l-1} \binom{n}{i} \frac{i!}{2i} p^i \\ &\leq \sum_{i=3}^{l-1} \frac{n^i}{2i} n^{i/l-i} = \sum_{i=3}^{l-1} \frac{n^{i/l}}{2i} = o(n)\end{aligned}$$

since $\binom{n}{i} \leq n^i/i!$. So by Markov's Inequality, $\mathbb{P}[X > \frac{n}{2}] = o(1)$, so there exists a graph with $\leq n/2$ cycles of length $< l$.

Now we make a quick remark:

Lemma 17. The chromatic number of G, χ_G, is $\geq \frac{n}{\alpha}$ where $\alpha =$ the size of the largest independent set.

If we can color using χ_G colors, then we can certainly form at least χ_G independent sets, which means the largest one must be at least n/χ_G.

Let a random variable $Y =$ the size of the largest independent set in G. Then by the union bound,

$$\begin{aligned}\mathbb{P}[Y \geq y] &\leq \binom{n}{y} (1-p)^{\binom{n}{2}} \\ &\leq n^y (e^{-p})^{\binom{n}{2}} \\ &\leq (ne^{-p(y-1)/2})^y \\ &= o(1) \quad \text{if we take } y = \frac{3}{p} \ln n = \frac{3n \ln n}{n^{1/l}}\end{aligned}$$

where we use $1 - x \leq e^{-x}$.

So by these two derivatiions, there exists a graph G s.t.

(a) G has $\leq n/2$ cycles of length l and

(b) G has a maximum independent set of size $\leq \frac{3}{p} \ln n$

as the probability any graph has either of these properties goes to 0. All that's left to do is to kill all of these cycles and independent sets.

Now get a new graph G' by removing from G one vertex on each cycle of length $< l$ so we remove $\leq n/2$ vertices. Then G' satisfies:

- $\geq n/2$ vertices
- girth $\geq l$
- chromatic number

$$\geq \frac{n/2}{3/p \ln n} = \frac{n/2}{3n \ln n / n^{1/l}} = \Omega(\frac{n^{1/l}}{\ln n}) \to \infty$$

So we can make the chromatic number as large as we like by taking n large enough. □

5.3 Monotone Circuits for Majority

The *Majority* function is such that

$$\text{Maj}(x_1, \ldots, x_n) = \begin{cases} 1 \text{ if } > n/2 \text{ of the } x_i = 1 \\ 0 \text{ if } < n/2 \text{ of the } x_i = 1 \end{cases}$$

(for our purposes, assume that n is odd). Using just AND or OR gates, we can create a circuit to represent this function. But this function is monotone, i.e. flipping input bits from 0 to 1 can't make the result flip from 1 to 0, so we actually don't need NOT gates to represent it. The question is how we would come up with such a construction that's efficient. Of course, we can use the probabilistic method to come up with one.

Theorem 18. There exists a monotone circuit that computes Maj using depth $O(\log n)$ and size $\text{poly}(n)$.

We care about this result, and monotone circuits in general, because of the following observations.

Fact 19. Almost all boolean functions $f : \{0,1\}^n \to \{0,1\}$ require circuits of size $2^n/2n$.

Proof Sketch. The proof is a simple counting argument. The number of functions f is 2^{2^n}, but the number of circuits of size $S \leq (16S^2)^S$, so if we take $S \sim 2^n/2n$ then $2^{2^n} \gg (16S^2)^S$. □

There's another very involved result that shows that CLIQUE requires superpolynomial circuit size for monotone circuits.

Proof of Maj. The proof is based on the Maj_3 graph. Construct a full ternary tree of depth t, and randomly connect three input values through a gate. Then we claim for $t = c \log n$,

$$\mathbb{P}[\text{circuit is correct on any fixed input } x_1, \ldots, x_n] \geq 1 - 2^{-n}.$$

The rest of the proof can be found in the notes. □

6 Thursday, February 9th

6.1 Methods of Conditional Probability

We can come up with a good bound for 3-SAT using the probabilistic method on conditional expectations.

6.2 Second Moment Method

Recall the following classic result:

Theorem 20 (Chebyshev's Inequality)**.** For any random variable X,

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \alpha] \leq \frac{\text{Var}(X)}{\alpha^2}.$$

While this doesn't seem like that strong of a bound, if all we know about X is it's variance and mean, then Chebyshev's inequality is indeed tight. If say X is the sum of i.i.d. random variables, we can achieve an exponential bound (Chernoff/Hoeffding). Sometimes we prefer to interpret this in terms of units of standard deviations, so we also look at it as

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \beta \text{Var}(X)] \leq \frac{1}{\beta^2}.$$

A common use case is in applications to random graphs $\mathcal{G}(n, p)$.

Example 6.1. Let $G \in \mathcal{G}(n, p)$. Does G contain a 4-clique?

Let X be the number of 4-cliques, and write it as $X = \sum_c X_c$ where X_c is 1 if the group of four vertices c is a clique and 0 otherwise. Then $\mathbb{P}[X_c = 1] = p^6$, so

$$\mathbb{E}[X] = \sum_c \mathbb{E}[X_c] = \binom{n}{4} p^6 = \Theta(n^4 p^6).$$

If $p \ll n^{-2/3}$, then $\mathbb{E}[X] \to 0$ as $n \to \infty$, and if $p \gg n^{-2/3}$, then $\mathbb{E}[X] \to \infty$ as $n \to \infty$.

But ideally we'd like to be able to say more, i.e. that with high probability the expectation goes to zero. So in other words, $\mathbb{P}[X > 0] = 0$ as $n \to \infty$ and $\mathbb{P}[X > 0] = 1$ respectively. If this is true, we say that $p(n)$ is a threshold function for containing a 4-clique.

Claim. $p(n) = n^{-2/3}$ is a threshold for 4-cliques.

Proof. Suppose $p \ll n^{-2/3}$. Then by Markov's inequality,

$$\mathbb{P}[X > 0] = \mathbb{P}[X \geq 1] \leq \mathbb{E}[X] \to 0$$

as $n \to \infty$.

Now suppose $p \gg n^{-2/3}$. We claim that $\frac{\text{Var}(X)}{\mathbb{E}[X]^2} \to 0$ as $n \to \infty$ (equivalently, $\mathbb{E}[X^2] = (1 + o(1))(E^2[X])$). Given this, we'd have

$$\mathbb{P}[X = 0] \leq \mathbb{P}[|X - \mathbb{E}[X]| \geq \mathbb{E}[X]] \leq \frac{\text{Var}(X)}{\mathbb{E}[X]^2} \to 0.$$

So let's prove out claim. First of all, we have that

$$\begin{aligned}
\text{Var}(X) &= \text{Var}(\sum_c X_c) \\
&= \sum_c \mathbb{E}[X_c^2] - \sum_c \mathbb{E}[X_c]^2 + \sum_{c \neq d} \mathbb{E}[X_c X_d] - \sum_{c \neq d} \mathbb{E}[X_c]\mathbb{E}[X_d] \\
&= \sum_c \text{Var}(X_c) + \sum_{c \neq d} \text{Cov}(X_c, X_d)
\end{aligned}$$

Now we just need to calculate the covariance $\text{Cov}(X_c, X_d)$. If $|C \cap D| \leq 1$, nothing affects each other (i.e. $\mathbb{E}[X_c X_d] = \mathbb{E}[X_c]\mathbb{E}[X_d]$) so $\text{Cov}(X_c, X_d) = 0$.

If $|C \cap D| = 2$, then the two sets share an edge. In order for all 6 vertices to be two 4-cliques, the 11 edges between them must be present, so $\mathbb{E}[X_c X_d] = p^{11}$. Upperbounding just this term (the $\mathbb{E}[X_c]\mathbb{E}[X_d]$ has negative contribution anyways) gives

$$\sum_{|C\cap D|=2} \mathbb{E}[X_c X_d] = \binom{n}{6}\binom{6}{2}\binom{4}{2}p^{11} = \Theta(n^6 p^{11}).$$

Finally, if $|C \cap D| = 3$, we perform a similar analysis to the last case, so

$$\sum_{|C\cap D|=3} \mathbb{E}[X_c X_d] = \Theta(n^5 p^9).$$

We don't check $|C \cap D| = 4$ since C, D are assumed to be distinct. Thus, we calculate

$$\frac{\text{Var}(X)}{\mathbb{E}[X]^2} \leq \frac{\Theta(n^4 p^6) + \Theta(n^6 p^{11}) + \Theta(n^5 p^9)}{\Theta(n^8 p^{12})} \to 0$$

assuming $p \gg n^{-2/3}$ □

This will serve as a sort of template for our future calculations.

Example 6.2. Let H be an arbitrary fixed graph. The threshold for G containing H exists and is $n^{-\alpha}$, where $\alpha = \min_{H'} v(H')/e(H')$ where H' is a sum over all induced subgraphs of H.

If H was a clique, then the ratio $n^{-v/e}$ couldn't get worse for arbitrary induced subgraphs, but in other cases it might; hence, the need to take a minimum.

One might ask what happens when our probability equals the threshold. For 4-cliques, when $p = cn^{-2/3}$, $X \sim \text{Poisson}(c^6/24)$. This proof isn't hard, but it just requires a steady hand. So $\mathbb{P}[G \text{ contains a 4-clique}] \to 1 - e^{-c^6/24}$. If you plot this as a function of c, you'll find that the graph changes smoothly from 0 to 1; we call this a "coarse" threshold.

For another example, we can consider the giant component problem. It turns out that the largest component in $\mathcal{G}(n, p)$ for $p = c/n$ is almost surely

- $\Theta(\log n)$ if $c < 1$
- $\Theta(n^{2/3}$ if $c = 1$
- $\Theta(n)$ is $c > 1$

The case where $p = 1/n$ is a "sharp" threshold for the property of having a component of size $\Theta(n)$.

Let's analyze $\mathcal{G}(n, p)$ with p constant. These will be *dense* random graphs, as the number of connected vartices is n times a constant, and the number of edges is n^2 times a constant.

Theorem 21. For $G \in \mathcal{G}(n, p)$ with $p \in (0, 1)$ constant, the size of a largest clique in G is $\sim 2\log_{1/p} n$ with probability 1 as $n \to \infty$.

There's a more refined result which says that the size of the largest clique is either $k(n)$ or $k(n)+1$. We won't prove this refined result, but in the proof of the above theorem we will limit it to a range of 6 or 7 numbers, from which it'll be clear how we would improve upon it.

From a computer science side, we have the following result with respect to finding such cliques. Unfortunately we're still off by a factor of 2.

Fact 22. Many polynomial time algorithms can find a clique of size $\geq (1-\epsilon)\log_2 n$ in $\mathcal{G}(n, 1/2)$ with very high probability.

We'll work through the proof next time, but instead we'll talk about the threshold.

Let $G \in \mathcal{G}(n, 1/2)$ and $X_k = \#$ of k-cliques in G. Then $\mathbb{E}[X_k] = \binom{n}{k} 2^{-\binom{k}{2}} = g(k)$ (not as p like usual). Then we'll find two values k_1 and k_2 so that $k_1 \leq k' \leq k_2$ where $g(k') \approx 0$. In other words, we will tightly bound where the graph of the expectation hits 0. Then by using second moment methods, we can also show that the graph behaves very close to the expectation in the long run.

It turns out that our threshold is $k_0 \sim 2 \log_2 n$. We can see this as

$$g(k) \sim \frac{n^k}{k!} 2^{-k^2/2} \approx 2^{k \log n - k^2/2 - k \log k}.$$

7 Tuesday, February 11th

We continue with our analysis of cliques in $\mathcal{G}(n,p)$, $p = 1/2$.

7.1 Large Cliques in Random Graphs

Theorem 23. The size of the largest clique in $\mathcal{G}(n,1/2)$ is $\sim 2\log_2 n$ with probability 1 as $n \to \infty$.

Proof. Let X_k be the number of cliques of size k, so that $\mathbb{E}[X_k] = \binom{n}{k}2^{-\binom{k}{2}} := g(k)$. Plotting $\log g(k)$ gives us a concave graph which intersects the k-axis at some point, which happens to be around $k_0 \sim 2\log n$ such that $g(k_0) = 1$.

To do this, let $k_2 = k_0 - c$ and $k_1 = k_0 + c$ be the two k's bounding the desired value, k_0. Then we must show the following.

(i) For $k = k_1 = k_0 + c$, $\mathbb{P}[X_{k_1} > 0] \to 0$ as $n \to \infty$.

(ii) For $k = k_2 = k_0 - c$, $\mathbb{P}[X_{k_2} > 0] \to 1$ as $n \to \infty$.

It is easy to see that in expectation, these statements hold, i.e. that

(i) $\mathbb{E}[X_{k_1}] \to 0$ as $n \to \infty$.

(ii) $\mathbb{E}[X_{k_2}] \to \infty$ as $n \to \infty$.

Also, if we look at the gradient of g, we find that

$$\frac{g(k+1)}{g(k)} = \frac{n-k}{k+1}2^{-k} \sim \frac{n}{2\log n}n^{-2} \to 0 \text{ as } n \to \infty.$$

This means that if we take a constant step, we will get the separation we need between k_1 and k_2. Now let's show these statements.

(i) By Markov's, $\mathbb{P}[X_{k_1} > 0] = \mathbb{P}[X_{k_1} \geq 1] \leq \mathbb{E}[X_{k_1}] \to 0$.

(ii) $\mathbb{P}[X_{k_2} = 0] \leq \mathbb{P}[|X_{k_2} - \mathbb{E}[X_{k_2}]| \geq \mathbb{E}[X_{k_2}]] \leq \dfrac{\operatorname{Var}(X_{k_2})}{(\mathbb{E}[X_{k_2}])^2}$, which we hope goes to 0.

We continue our analysis of the second part. To make notation cleaner, let $X := X_{k_2}$ and write X as the sum of indicators $X = \sum_S X_S$ where $X_S = 1$ if S is a clique and 0 otherwise. Then

$$\begin{aligned}
\operatorname{Var}(X) &= \sum_S \operatorname{Var}(X_S) + \sum_{S\sim T} \operatorname{Cov}(X_S, X_T) \\
&\leq \sum_S \mathbb{E}[X_S^2] + \sum_{S\sim T} \mathbb{E}[X_S X_T] \\
&\leq \sum_S \mathbb{E}[X_S] + \sum_{S\sim T} \mathbb{E}[X_S X_T] \\
&\leq \mathbb{E}[X] + \sum_{S\sim T} \mathbb{E}[X_S X_T]
\end{aligned}$$

where we assume S, T have some intersection as cliques, otherwise the covariance would just be zero.

So that

$$\frac{\operatorname{Var}(X)}{\mathbb{E}[X]} = \frac{1}{\mathbb{E}[X]} + \frac{1}{\mathbb{E}[X]^2}\sum_{S\sim T}\mathbb{E}[X_S X_T].$$

We can drop the first expectation since it goes to 0, so we'll focus on the sum on the right. So,

$$\frac{1}{\mathbb{E}[X]^2}\sum_{S\sim T}\mathbb{E}[X_S X_T] = \frac{1}{\mathbb{E}[X]^2}\sum_{S}\mathbb{E}[X_S]\sum_{T\sim S_0}\mathbb{P}[X_T = 1 | X_{S_0} = 1] = \frac{\sum_{T\sim S_0}\mathbb{P}[X_T = 1|X_{S_0}=1]}{\mathbb{E}[X]}.$$

We break down the summation very carefully now.

$$\begin{aligned}\frac{1}{\mathbb{E}[X]} = \sum_{T\sim S_0}\mathbb{P}[X_T = 1|X_{S_0}=1] &= \sum_{i=2}^{k_2-1}\frac{\binom{k_2}{i}\binom{n-k_2}{k_2-1}2^{-[\binom{k_2}{2}-\binom{i}{2}]}}{\binom{n}{k_2}2^{-\binom{k_2}{2}}} \\ &= \sum_{i=2}^{k_2-1}\frac{\binom{k_2}{i}\binom{n-k_2}{k_2-1}2^{\binom{i}{2}}}{\binom{n}{k_2}} \\ &= \sum_{i=2}^{k_2-1} f(i) \le k_2 \max_i f(i)\end{aligned}$$

It turns out that for large enough constant c, one can force $f(i)$ to achieve its maximum at the left hand most point of the interval, aka $\arg\max f(i) = 2$. So

$$\cdots \le k_2\frac{\binom{k_2}{2}\binom{n-k_2}{k_2-1}2}{\binom{n}{k_2}} \le \frac{k_2\times k_2^2\times k_2^2}{n^2} = \Theta(\log^5 n/n^2)\to 0.$$

□

7.2 Random k-SAT

Instead of looking at random graphs, we can also look at random k-SAT.

The input is a boolean formula φ in k-CNF, i.e. $(x_1\vee x_2\wedge\cdots\wedge\bar{x_k})\vee(\dots)\dots$. The output is if φ is satisfiable.

Definition 24 (Random k-SAT). $\varphi_k(r, r_n)$ denotes a random k-CNF formula with n variables and r_n clauses, which we call the *density*.

We conjecture that for every $k \ge 2$, $\exists$ a threshold value r_k^* such that

(i) $r > r_k^* \implies \mathbb{P}[\varphi_k(n, r_n) \text{ satisfiable}] \to 0$ as $n\to\infty$.

(ii) $r < r_k^* \implies \mathbb{P}[\varphi_k(n, r_n) \text{ satisfiable}] \to 1$ as $n\to\infty$.

This has been proved for $k = 2 (r_2^* = 1)$, and for all sufficiently large k, for which $r_k^* = 2^k\ln 2 - \frac{1}{2}(1+\ln 2) - \epsilon_k$, where $\epsilon_k\to 0$ as $k\to\infty$. While the $k=3$ case is still open, theoretical bounds state that $3.52 \le r_3^* \le 4.42$ while experiments suggest that the true value lies around 3.

Proving this conjecture for large k was a monumental effort that concluded in 2014 after 30 years of effort.

We can prove the $k = 2$ case from first principles, so we will go ahead and do so.

Theorem 25. $r_2^* = 1$ for random 2-SAT.

Proof. (i) Take a random 2-SAT formula φ with $m = (1-\epsilon)n$ clauses. We claim that $\mathbb{P}[\varphi \text{ satisf.}]\to 1$ as $n\to\infty$. For this, we use the typical graph representation, sometimes

called a "graph of implications," which has $2n$ vertices, one for every variable and its negation, and arrows if one variable being true implies another. From this, we see that

$$\varphi \text{ is satisfiable } \iff G_\varphi \text{ does not contain a "contradictory" cycle.}$$

A *bicycle* is a path $u, w_1, w_2, \ldots, w_k, v$ (different k than k-SAT) in G_φ, where the w_i are literals on *distinct* variables and $u, v \in \{w_i, \bar{w}_i | 1 \le i \le k\}$.

Claim. G_φ containing no bicycle means that it contains no contradictory cycle.

It's clear that a bicycle is much stronger than a contradictory 2-cycle, as any such bicycle can get shrunk to the offending 2-cycle. But the extra structure of a bicycle, especially that all the variables are distinct, will be helpful in our proof.

$$\begin{aligned}
\mathbb{P}[\varphi \text{ satisf.}] &\le \mathbb{P}[G_\varphi \text{ contains a bicycle}] \\
&\le \sum_{k=2}^{n} n^k 2^k (2k)^2 m^{k+1} \left(\frac{1}{4\binom{n}{2}}\right)^{k+1} \\
&= \frac{2}{n} \sum_{k=2}^{n} k^2 \left(\frac{m}{n-1}\right)^{k+1} \\
&= \frac{2}{n} \sum_{k=2}^{n} k^2 \left((1-\epsilon)\frac{n}{n-1}\right)^{k+1} \to 0 \text{ as } n \to infty
\end{aligned}$$

as the summation is a geometric series (polynomial term being negligible) so the entire expression is dominated by the $2/n$.

(ii) Take $m = (1+\epsilon)n$. We claim that $\mathbb{P}[\varphi \text{ satisf.}] \to 0$ as $n \to \infty$.

A *snake* is a sequence of literals $w_1, w_2, \ldots, w_s$ with *distinct* variables, where $s = 2t - 1$ is odd. Associate with any snake A a set of clauses

$$F_A := (w_t \vee w_1) \wedge (\bar{w}_1 \wedge w_2) \vee (\bar{w}_2 \vee w_3) \wedge \cdots \wedge (w_{s-1} \vee w_s) \wedge (\bar{w}_s \vee \bar{w}_t).$$

We claim that $\mathbb{P}[\varphi \text{ contains } F_A \text{ for some snake } A] \to 1$.

Let $X_A = 1$ if φ contains every clause in F_A exactly once, and 0 otherwise. Then $X = \sum_A X_A$. It suffices to prove that for any fixed A,

$$\frac{1}{\mathbb{E}[X]} \sum_{B \sim A} \mathbb{P}[X_B = 1 | X_A = 1] \to 0 \text{ as } n \to \infty.$$

At the beginning of next time, we'll carry out these computations.

□

8 Thursday, February 13th

8.1 Finishing Random k-SAT Formula

Today we will finish proving Theorem 25. Recall from last time that we proved for $r \leq (1-\epsilon)$, $\mathbb{P}[\varphi \text{ satisf.}] \to 1$. Now we will show that for $r \geq (1+\epsilon)$, $\mathbb{P}[\varphi \text{ satisf.}] \to 0$.

A *snake* is a sequence of literals $w_1, w_2, \ldots, w_s$ with *distinct* variables, where $s = 2t-1$ is odd. Associate with any snake A a set of clauses

$$F_A := (w_t \vee w_1) \wedge (\bar{w}_1 \wedge w_2) \vee (\bar{w}_2 \vee w_3) \wedge \cdots \wedge (w_{s-1} \vee w_s) \wedge (\bar{w}_s \vee \bar{w}_t).$$

We claim that $\mathbb{P}[\varphi \text{ contains } F_A \text{ for some snake } A] \to 1$.

Set $X_A = 1$ if φ contains every clause in F_A exactly once, and 0 otherwise. Then $X = \sum_A X_A$. Our claim is equivalently that $\mathbb{P}[X > 0] \to 1$.

Based on our previous experience with the second moment method, it suffices to prove that the covariances become dominated by the expectations, i.e. for any fixed A,

$$\frac{1}{\mathbb{E}[X]} \sum_{B \sim A} \mathbb{P}[X_B = 1 | X_A = 1] \to 0 \text{ as } n \to \infty.$$

First we write down the probability that a specific bunch of clauses shows up, i.e.

$$\mathbb{E}[X_A] = \binom{m}{2t} (2t)^1 \left(\frac{1}{4\binom{n}{2}}\right)^{2t} \left(1 - \frac{2t}{4\binom{n}{2}}\right)^{m-2t} =: h(2t).$$

In other words, $h(2t)$ is the probability of some snake with $2t-1$ distinct variables. Thus,

$$\mathbb{P}[X_B = 1 | X_A = 1] = \frac{h(4t-k)}{h(2t)}$$

where k is the size of the clusters of A, B. Then,

$$\frac{1}{\mathbb{E}[X]} \sum_{B \sim A} \mathbb{P}[X_B = 1 | X_A = 1] = \sum_{k=1}^{2t-1} \frac{h(4t-k)}{h(2t)} p_k$$

where $p_k =$ the probability random k snakes containing k of the same clauses with A. This calculation of p_k is routine combinatorics, so we will omit it.

Analyzing our function h, we see that

$$h(z) = (1 + o(1)) \left(\frac{m}{2n(n-1)}\right)^z$$

provided that $z \ll \sqrt{n}$. So our sum from above is now

$$(1 + o(1)) \sum_{k=1}^{2t-1} \left(\frac{2n(n-1)}{m}\right)^k p_k$$

Now we're pulling out some tricks that you defnitely won't follow We have the following two facts:

(a) $p_k \leq ctn \left(\frac{1}{2n}\right)^k$ for $1 \leq k \leq 2t$.

(b) $p_k \leq \frac{c't^9}{n} \left(\frac{1}{2n}\right)^k$ for $1 \leq k \leq t-1$.

Using these, we can split our sum into two chunks. We see that

$$\sum_{k=1}^{t-1}\left(\frac{2n(n-1)}{m}\right)^k p_k \le \frac{c't^9}{n}\sum_{k=1}^{t-1}\left(\frac{n-1}{m}\right)^k \to 0$$

since $m \ge (1+\epsilon)n$ and taking $t = n^{1/10}$. Then we also have

$$\sum_{k=t}^{2t-1}\left(\frac{2n(n-1)}{m}\right)^k p_k \le ctn\sum_{k=t}^{2t}\left(\frac{n-1}{m}\right)^k \to 0$$

since $\frac{n-1}{m} \le 1-\delta$ and first term is $(1-\delta)^{n^{1/10}}$.

Now we'll show an easy upper bound $r_k^* \le 2^k \ln 2$ for arbitrary k-SAT.

Let

$$X_\sigma = \begin{cases} 1 \text{ if assigment } \sigma \text{ satisf.} \varphi \\ 0 \text{ o.w.} \end{cases}$$

and let $X = \sum X_\sigma$. φ satisfiable is equivalent to showing that $X > 0$. Using first moment methods, we find that

$$\mathbb{E}[X] = 2^n(1-2^{-k})^{rn} \to 0$$

as $n \to \infty$ for $r > 2^k \ln 2$. So by Markov's inequality, $\mathbb{P}[X > 0] \to 0$ as $n \to \infty$.

Next is to show the lower bound with second moment methods. We first state the following classical claim.

Claim. For any non-negative r.v. X,

$$\mathbb{P}[X > 0] \ge \frac{(\mathbb{E}[X])^2}{\mathbb{E}[X^2]}.$$

Proof. Exercise. Use Cauchy-Schwarz on Chebyshev. □

It is insightful to notice that traditional Chebyshev gives us $\mathbb{P}[X > 0] \ge 2 - \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2}$, which is not useful for us as this is trivial sometimes.

By Friedgut (threshold result for random k-SAT), it suffices to prove that $\mathbb{P}[X > 0] \ge \delta > 0$ since then $\mathbb{P}[X > 0] \to 1$.

Computing second moments now gives

$$\begin{aligned}\mathbb{E}[X^2] = \sum_{\sigma,\tau}\mathbb{E}[X_\sigma X_\tau] &= \sum_{\sigma,\tau}\mathbb{P}[\text{ both } \sigma,\tau \text{ satisfy } \varphi] \\ &= \sum_{\sigma,\tau}\prod_{i=1}^{m}\mathbb{P}[\text{both } \sigma,\tau \text{ satisfy } c_i]\end{aligned}$$

This reduces oru calculation to just checking if two arbitrary assignments satisfy a clause.

$$\mathbb{P}[\sigma,\tau \text{ both satisfy } c_i] = 1 - 2^{1-k} + 2^{-k}\alpha^k =: f(\alpha)$$

where $z = \alpha n$ is the *overlap* of σ, τ. Thus,

$$\mathbb{E}[X^2] = 2^n\sum_{z=0}^{n}\binom{n}{z}f\left(\frac{z}{n}\right)^m$$

Now we'll try approximating this binomial coefficient (in doing so, making approximation and ignoring issues about integers, etc.).

$$\binom{n}{\alpha n} \sim \left(\frac{1}{\alpha^\alpha (1-\alpha)^{1-\alpha}}\right)^n \Theta\left(\frac{1}{\sqrt{n}}\right).$$

Plugging this in gives

$$\mathbb{E}[X^2] = \left(\frac{2f(\alpha)^r}{\alpha^\alpha (1-\alpha)^{1-\alpha}}\right)^n \Theta\left(\frac{1}{\sqrt{n}}\right)$$

and

$$\mathbb{E}[X]^2 = [2^n(1-2^{-k})^{rn}]^2 = (4f(1/2)^r)^n$$

where we plug in $f(1/2)$ instead for the expression to make it easier to compare with $\mathbb{E}[X^2]$. Putting this together gives

$$\frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2} = \sum_\alpha \left(\frac{\Lambda(\alpha)}{\Lambda(1/2)}\right)^n \quad \text{where } \Lambda(\alpha) = \frac{2f(\alpha)^r}{\alpha^\alpha (1-\alpha)^{1-\alpha}}.$$

We want to show that this geometric sum is upper bounded by some M as $n \to \infty$. One hope would be that Λ is maximized at $1/2$, so that oour terms would definitely be bounded. But unfortunately this isn't the case; the probability of satisfying a clause given more overlap in the assignments becomes higher, so $f(\alpha)$ is always increasing, while the bottom (essentially the entropy function) maximizes at $\alpha = 1/2$.

The solution here is to modify our random variable to reweight the correlations. Intuitively this works because we haven't taken advantage of any structure in our random variable, all we care is that it goes to 0 as $n \to \infty$.

9 Tuesday, February 18th

9.1 Pairwise Independent RVs

We first define what it means for RVs to be pairwise independent.

Definition 26. A collection $\{X_i\}_{i=1}^n$ of r.v.'s is (k-wise) *pairwise independent* if $\forall\, i \neq j$, and all a, b,

$$\mathbb{P}[X_i = a \wedge X_j = b] = \mathbb{P}[X_i = a] \times \mathbb{P}[X_j = b]$$

$$\mathbb{P}\left[\bigwedge_{i \in S} X_i = a_i\right] = \prod_{i \in S} \mathbb{P}[X_i = a_i] \quad \forall S \subset \{1, \ldots, n\}, |S| \leq k.$$

One application of having pairwise independent RVs is creating a probability amplification of Monte Carlo algorithms on a language L. Usually, we'd have a poly time algorithm $\mathcal{A}$ using m random bits s.t.

$$x \in L : \mathbb{P}[\mathcal{A}(x) \text{ outputs "yes"}] \geq 1/2$$

$$x \notin L : \mathbb{P}[\mathcal{A}(x) \text{ outputs "yes"}] = 0$$

Suppose we want error probability $\leq 1/r$. We can achieve this using $O(\log r)$ repeated independent trials, which uses $O(m \log r)$ random bits. But this is rather intensive int erms of randomness; as we know, generating random bits is rather intensive.

With pairwise independence, to achieve an error probability of $1/r$ ($r \leq 2^m$), we claim that this requires just $2m$ random bits in running time $O(rm)$.

Assume $x \in L$, so $\mathbb{P}[\text{outputs "yes"}] = p \geq 1/2$. Identify executions with strings over $\{0,1\}^m$. Pick r pairwise independent uniform random samples from $\{0,1\}^m$. Let

$$X_i = \begin{cases} 1 & \text{if } A(x) \text{ outputs "yes" on executation seq. } i \\ 0 & \text{otherwise} \end{cases}$$

and let $X = \sum_{i=1}^r X_i$. Then

$$\mathbb{P}[X = 0] \leq \mathbb{P}[|X - \mathbb{E}[X]| \geq \mathbb{E}[X]] \leq \frac{\text{Var}(X)}{\mathbb{E}[X]^2}.$$

By pairwise independence, $\text{Var}(X) = \sum \text{Var}(X_i) = rp(1-p)$. Then

$$\mathbb{P}[X = 0] \leq \frac{rp(1-p)}{(rp)^2} = \frac{1-p}{p} \times \frac{1}{r} \leq \frac{1}{r}$$

since $p \geq 1/2$.

9.1.1 Generating Pairwise Independent RVs

Choose prime q s.t. $2^m < q < 2^{m+1}$, and let us work over $\mathbb{Z}_q$. Pick a, b independently and u.a.r. over $\mathbb{Z}_q$. Define $f_{a,b}(x) = ax + b \pmod q$.

Claim. The family $\{f_{a,b}(x) : x \in \mathbb{Z}_q\}$ is pairwise independent and uniform.

Proof. Consider

$$\mathbb{P}_{a,b}[ax_1 + b = z_1 \wedge ax_2 + b = z_2] = \frac{1}{q^2}$$

which is a pair of linear equations over $\mathbb{Z}_q$ in a, b. There is also always a unique solution for a, b for any choices of z_1, z_2. □

To get d-wise independent random variables over $\mathbb{Z}_q$, the family

$$\left\{\sum_{i=1}^{d-1} a_i x^i : x \in \mathbb{Z}_q\right\}$$

suffices for independent random $\{a_i\}_{i=0}^{d-1} \in \mathbb{Z}_q$.

There are other constructions of pairwise independent random variables, such as the Toeplitz matrix.

9.2 Ramsey Theory

We're going back to Ramsey Theory to see how saving on random bits can help (and in certain cases, we can actually just enumerate *all* possibilities, removing the need to use randomness at all!).

Randomly color edges of K_n with red/blue, and let $X = \#$ of monochromatic k-cliques. Then

$$\mathbb{E}[X] = \binom{n}{k} 2^{-\binom{k}{2}+1} =: f(k).$$

For $n = 2^{k/2}$, $f(k) < 1 \implies$ there exists coloring with no monochromatic k-clique. This lends itself to being parallelizable.

One observation to note is that we only need independence among all of the k edges in a clique. Set $r = \binom{n}{2}$ and $d = \binom{k}{2}$. Then having d-wise independent r.v.'s for colors are OK.

Construct a d-wise independent family over $\mathbb{Z}_q$ where $q > r$. The size of the sample space is $q^d \sim \binom{n}{2}^{\binom{k}{2}} = n^{O(k^2)}$, so we can try all sample points in poly(n) time (even in parallel), which is $O(\log n)$ in parallel time.

9.3 Universal Hashing

We have a large universe of possible keys U, and we have a relatively small database S within this universe. We want to create a hash table T for values on these keys (think dictionaries) that offers basic operations like Add, Delete, and Search.

For our purposes, let $|U| = M \gg |S| = n$, and ideally $|T| = O(|S|)$ (we wouldn't want a hash table much larger than the size of our database). We can't make entirely random hash functions however, as the set of all functions $f : U \to T$ is size n^M, which would require $O(M \log n)$ bits.

Definition 27. A family $\mathcal{H}$ of hash functions $U \to T$ is *2-universal* if $\forall x \neq y \in U$,

$$\mathbb{P}_{h \in \mathcal{H}}[h(x) = h(y)] \leq \frac{1}{n}.$$

We can achieve this using pairwise independent family, requiring $O(\log M)$ random bits. For any x,

$$\mathbb{E}[\#\text{ of collisions with } x] \leq (|S| - 1)\frac{1}{n} \leq 1.$$

Can we derandomize this? In other words, can we accurately calculate the number of collisions in T? It turns out we can, and the technique is called *double hashing* due to TKS.

The idea is as follows. You are trying to setup this hash function once, but you want it to have (for sure) very few collisions. There is a way to do this in the setup so that you use randomness to create a hash functions, and if it's good you keep it, otherwise you toss it. In expectation however, you'll only have to do this twice, maybe four times.

There is one caveat: the hash function is for a static dictionary, but it'll fall apart if we start inserting elements.

The key idea is to have an intermediate hash table T linking to our final hash table T'. We keep collisions that happen in T, but then resolve collisions in T when we map over to T' with mini-hash functions. We first make a few observations.

Claim. If we use a 2-universal family of hash functions to map a set of size b to a table of size b^2, then $\mathbb{P}[\exists \text{ a collision}] \leq 1/2$.

Proof. Let $X =$ the number of collisions. Then $\mathbb{E}[X] = \binom{b}{2} \times \frac{1}{b^2} \leq \frac{1}{2}$. Finish with Markov. □

We will use this result to construct hash functions from T to T' that resolve collisions in T by mapping to sets of size that are quadratic in the expected number of collisions.

For the next claim, let b_i be the expected bin size of the target of every hash function, which also corresponds to the number of elements that map to each i.

Claim. If we use a 2-universal family to map S to a table of size $n \geq |S|$ then $\mathbb{P}[\sum b_i^2 \geq 4|S|] \leq 1/2$.

In other words, accounting for collisions, the expected result will take up space linear in the size of our original set.

Proof. Let $X =$ the number of collisions again. Then

$$\mathbb{E}[X] = \sum_i \binom{b_i}{2} = \frac{1}{2}\sum (b_i^2 - b_i) = \frac{1}{2}\left(\sum b_i^2 - |S|\right).$$

Isolating the summation, we compute

$$\begin{aligned}\mathbb{E}\left[\sum b_i^2\right] &= 2 \times \mathbb{E}[X] + |S| \\ &\leq 2\binom{|S|}{2}\frac{1}{n} + |S| \\ &\leq 2|S|\end{aligned}$$

from which we finish with Markov. □

We should think about the number of bits required to represent each of the mini-hash functions mapping our collisions from T to T', but this only needs

$$\sum b_i \log b_i = O\left(\sum b_i^2\right) = O(|S|) \text{ bits},$$

which is linear in the size of our original set.

10 Thursday, February 20th

10.1 Unbiased Estimators

We have a ground set U and an interesting set $A \subseteq U$. Our goal is to estimate $|A|$ (or $|A|/|U|$). An *unbiased estimator* X is one for which $\mathbb{E}[X]$ equals the quantity we are trying to estimate. For our purposes, if we let $X_i = 1$ if $i \in A$ and $X = \frac{1}{t}\sum_{i=1}^{t} X_i$, then $\mathbb{E}[X]$ is an unbiased estimator of $|A|/|U|$.

The only room for improvement in this setting is to increase t and hope that we get a better bound in that case.

Theorem 28. Let $\{X_i\}_{i=1}^{t}$ be i.i.d. r.v.'s with $\mathbb{E}[X_i] = \mu$ and $\mathrm{Var}(X_i) = \sigma^2$. Let $X = \frac{1}{t}\sum X_i$. Then if $t > \frac{4}{\epsilon^2} \cdot \frac{\sigma^2}{\mu^2}$ we have

$$\mathbb{P}[|X - \mu| > \epsilon\mu] \leq \frac{1}{4}.$$

There's two important takeaways from this theorem. If we want to be within $\epsilon\mu$ of the mean, then we need to pay a price of $1/\epsilon^2$ for that accuracy. Second, if our interesting set A is much smaller than the ground set, we also pay a price for trying to sample it better. We can see this as $\mathbb{E}[X_i] = p$ and $\mathrm{Var}(X_i) = p(1-p) \sim p$, so

$$\frac{\sigma^2}{\mu^2} = \frac{p}{p^2} = \frac{1}{p}.$$

Proof. We have $\mathbb{E}[X] = \mu$ and $\mathrm{Var}(X) = \frac{1}{t}\sigma^2$. Apply Chebyshev's to get

$$\mathbb{P}[|X - \mu| > \epsilon\mu] \leq \frac{\mathrm{Var}(X)}{\epsilon^2\mu^2} = \frac{\sigma^2}{t\epsilon^2\mu^2}.$$

□

Claim. Taking the *median* of t' such trials gives an estimate in the range $(1\pm\epsilon)\mu$ with probability $\geq 1 - \delta$ for $t' \geq c\log(1/\delta)$ (similar argument as in lecture 1).

10.2 Counting Problems

We have some different classes of decision problems (where we count instead of just decide on existence) due to Valiant.

#P: A function $f : \sigma^* \to \mathbb{N}$ is in #P if $\exists$ a polynomiall time nondeteterministic TM M s.t. $\forall x \in \sigma^*$ the number of accepting computations of M on x is $f(x)$.

#P-completeness: $f \in \#P$ is $\#P$-complete if every other $f' \in \#P$ can be reduced to f in polynomial time.

Fact 29. There are many important problems whose decision version is in P but whose counting version is $\#P$-complete.

Some examples are perfect matchings (equivalent to permanent).

11 Tuesday, February 25th

I was absent. That is all.

12 Thursday, February 27th

12.1 Approximating the Permanent

For any 0-1 matrix A, the permanent of A is

$$\text{per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}.$$

If A is an adjacency matrix, then $\text{per}(A)$ is the number of matchings in G_n.

There is an algorithm by Ramussen which give a (random) unbiased estimator for the permanent, which is as follows. If $n = 0$, then $X_A = 1$. Otherwise let $W_A := \{j : a_{1j} = 1\}$. Pick $i \in W_A$ u.a.r., and output $|W_A| \times X_{A_{1,j}}$.

Theorem 30. Let A be a random $n \times n$ 0-1 matrix, $\omega(n)$ be any function s.t. $\omega(n) \to \infty$, and let X_A be this unbiased estimator for the permanent. Then

$$\mathbb{P}\left[\frac{\mathbb{E}[X_A^2]}{\mathbb{E}[X_A]^2} > n \cdot \omega(n)\right] \to 0 \text{ as } n \to \infty.$$

Corollary 31. Algorithm based on X_A is an f.p.r.a.s. for $\text{per}(A)$ with probability $\to 1$ over choice of A with $\Theta(\frac{n \cdot \omega(n)}{\epsilon^2})$ repeated trials

Total running time therefore is $O(n^3 \cdot \omega(n)/\epsilon^2)$.

Forewarning: there will be two probability spaces we refer to throughout in this proof. There is $\mathcal{A}_n$, the space of random $n \times n$ 0-1 matrices with uniform distribution, so $A \in \mathcal{A}_n$. We will write $\mathbb{E}_{\mathcal{A}_n}$ when we take expectations w.r.t. this space, and likewise for probabilities.

Proof. We first make a claim.

Claim. (a) $\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A]] = \frac{n!}{2^n}$.

(b) $\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A^2] = \frac{1}{4^n}\prod_{i=1}^{n}(i^2 + i)$.

The first claim is easy to see, as each choice has probability $1/2$, among all $n!$ possibilities. The second one not so much, so we construct such a proof.

Proof of Claim. Let $W_i \sim \text{Bin}(i, 1/2)$. Then $\mathbb{E}[W_i] = i/2$ and $\mathbb{E}[W_i^2] = \frac{i^2+i}{4}$. So

$$\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A]] = \prod_{i=1}^{n} \mathbb{E}[W_i] = \prod_{i=1}^{n} \frac{i}{2} = \frac{n!}{2^n}$$

and

$$\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A^2]] = \prod_{i=1}^{n} \mathbb{E}[W_i^2] = \prod_{i=1}^{n} \frac{i^2+i}{4} = \frac{1}{4^n}\prod_{i=1}^{n}(i^2 + i).$$

□

Given this, one corollary is that

$$\frac{\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A]]}{\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A^2]]} = \frac{\frac{1}{4^n}\prod(i^2+i)}{\frac{1}{4^n}\prod i^2} = \prod_{i=1}^{n}\left(\frac{i+1}{i}\right) = n+1$$

as the product telescopes.

Lemma 32 (Main Lemma). For any $\omega(n) \to \infty$, $\mathbb{P}_{\mathcal{A}_n}\left[\text{per}(A) < \frac{\mu(n)}{\omega(n)}\right] \to 0$ as $n \to \infty$ where $\mu(n) = \mathbb{E}_{\mathcal{A}_n}[\text{per}(A)] = n!/2^n$.

We need this lemma to get a good lower bound on how close our indicator is to the permanent. Markov's tells us that the values can't get too much greater than the mean, but nothing about the other direction. First let's see how this proves our original goal.

Proof of Main Theorem. By Markov,

$$\mathbb{P}_{\mathcal{A}_n}[\mathbb{E}[X_A^2] \geq \omega(n)\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A^2]]] \leq \frac{1}{\omega(n)} \to 0.$$

By the Main Lemma,

$$\mathbb{P}_{\mathcal{A}_n}\left[\frac{1}{\mathbb{E}[X_A]^2} > \frac{\omega(n)^2}{\mathbb{E}_{\mathcal{A}_n}[\mathbb{E}[X_A]^2}\right] \to 0.$$

So our afore mentioned corollary, the ratios are well behaved, so comparing the ratios of the events gives

$$\mathbb{P}_{\mathcal{A}_n}\left[\frac{\mathbb{E}[X_A^2]}{\mathbb{E}[X_A]^2} > (n+1)\omega^3(n)\right] \to 0.$$

□

To finish, we refine our probability space. Let $\mathcal{A}_{n,m}$ = uniformly random 0-1 $n \times n$ matrices with exactly m 1's. Pick $M \sim \text{Bin}(n^2, 1/2)$ and pick $A \in \mathcal{A}_{n,M}$.

Lemma 33 (Main' Lemma)**.** Suppose $m = m(n)$ satisfies $\frac{m^2}{n^3} \to \infty$ as $n \to \infty$. Then,

(a)
$$\mathbb{E}_{\mathcal{A}_n,m}[\text{per}(A)] = n!\left(\frac{m}{n^2}\right)^n \exp\left(-\frac{n^2}{2m} + \frac{1}{2} + O\left(\frac{n^3}{m^2}\right)\right)$$

(b)
$$\frac{\mathbb{E}_{\mathcal{A}_n,m}[\text{per}(A)^2]}{(\mathbb{E}_{Acal_n,m}[\text{per}(A)])^2} = 1 + O\left(\frac{n^3}{m^2}\right).$$

First we see how this lemma implies our Main Lemma.

Proof that Main' $\implies$ Main. Can assume (w. prob $\to 1$) that *both* $M \geq \frac{n^2}{2} - n \cdot \omega'(n)$ and

$$\mathbb{P}_{\mathcal{A}_{n,m}}\left[\text{per}(A) < \frac{1}{2}\mathbb{E}_{\mathcal{A}_{n,m}}(\text{per}(A))\right] < \frac{4\text{Var}}{\mathbb{E}^2} \to 0$$

by part (b). Then with probability 1,

$$\begin{aligned}
\frac{\text{per}(A)}{\mu(n)} &\geq \frac{\frac{1}{2}\mu(n, \frac{n^2}{2} - \omega' n)}{\mu(n)} \\
&= \frac{1}{2} \cdot \frac{2^n}{n!} \cdot n! \cdot \left(\frac{n^2/2 - \omega' n}{n^2}\right)^n \exp\left\{-\frac{n^2}{n^2 - 2\omega' n} + \frac{1}{2} + OO(1/n)\right\} \\
&\geq \frac{1}{2}\left(1 - \frac{2\omega'}{n}\right)^n \exp(-1 + O(1/n)) \\
&\sim \frac{1}{2}\exp(-2\omega' - 1)
\end{aligned}$$

but this is definitely greater than $1/\omega(n)$ since ω' is your favorite slowly-growing function. □

Now all that's left is to prove this Main' Lemma.

Proof of Main' Lemma. Recall $\text{per}(A) = \#$ of PMs in G_A. Let H be any subgraph with t edges. Then

$$\mathbb{P}[H \text{ is a subgraph of } G_A] = \frac{\binom{n^2-t}{m-t}}{\binom{n^2}{m}} =: q(t).$$

One can check that

$$q(t) = \frac{m(m-1)\dots(m-t+1)}{n^2(n^2-1)\dots(n^2-t+1)} = \left(\frac{m}{n^2}\right)^t \exp\left\{-\frac{t^2}{2}\left(\frac{1}{m} - \frac{1}{n^2} + O(\frac{n^3}{m^2})\right)\right\}$$

assuming that $t = O(n)$. We get this by factoring out m/n^2 from each term carefully, and approximating each $(1 - i/m)$ term with $\exp(-i/m)$, and similarly for the n^2's.

Thus,

$$\begin{aligned}
\mathbb{E}_{\mathcal{A}_{n,m}}[\text{per}(A)] &= \sum_{\text{PM's } H} \mathbb{P}[H \text{ subgraph of } G_A] \\
&= n!q(n) \\
&= n!\left(\frac{m}{n^2}\right)^n \exp\left\{-\frac{n^2}{2m} + \frac{1}{2} + O\left(\frac{n^3}{m^2}\right)\right\}
\end{aligned}$$

as desired.

To prove the second part, we need to look at pairs of matchings as one usually does in these types of proofs. We see that

$$\begin{aligned}
\mathbb{E}_{\mathcal{A}_{n,m}}[\text{per}(A)^2] &= \sum_{H,H' \text{ PMs}} \mathbb{P}[H, H' \text{ are subgraphs of } G_a] \\
&= \sum_{k=0}^{n} n!\binom{n}{k} D(n-k)q(2n-k)
\end{aligned}$$

where we can use $D(n-k) \sim \frac{(n-k)!}{e}$ as an asymptotic bound. The rest of the computation is delegated to the notes. □

□

13 Tuesday, March 3rd

Now we're moving on to using higher moment bounds in our analyses. However, in order to do so, we need to enforce more structure into our problems in order to extract out these higher moments and use their corresponding bounds.

13.1 Chernoff/Hoeffding Bounds

Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables where $\mathbb{E}[X_i] = \mu_i$ and $\text{Var}(X_i) = \sigma_i^2$, and let $X = \sum_{i=1}^n X_i$ be their sum.

A standard computation shows that $\mu = \mathbb{E}[X] = \sum \mu_i = n\mu_1$ and $\sigma^2 = \text{Var}(X) = n\sigma_1^2$.

Before looking at the Chernoff Bound, we first observe some other bounds. Using the Central Limit Theorem, $(X - \mu)/\sigma \sim \mathcal{N}(0, 1)$, so

$$\mathbb{P}[|x - \mu| > \beta\sigma] \to \frac{1}{\sqrt{2\pi}} \int_\beta^\infty e^{-t^2/2} dt \sim \frac{1}{\sqrt{2\pi}} e^{-\beta^2/2}.$$

However, this doesn't give us any guarantee on the rate of convergence, which is what we'd really liike to know. Instead, we work from first principles. We will first prove this tail bound result, which will lead to all of our other results.

Theorem 34. Let $X_1, \ldots, X_n$ be independent 0-1 random variables with $\mathbb{E}[X_i] = p_i$. Let $X = \sum_{i=1}^n X_i$, $\mathbb{E}[X] = \mu = \sum_{i=1}^n p_i$, $p = \frac{\mu}{n}$. Then

1. $\mathbb{P}[X \geq \mu + \lambda] \leq \exp\left\{-nH_p\left(p + \frac{\lambda}{n}\right)\right\}$ for $0 < \lambda < n - \mu$.
2. $\mathbb{P}[X \leq \mu - \lambda] \leq \exp\left\{-nH_{1-p}\left(1 - p + \frac{\lambda}{n}\right)\right\}$ for $0 < \lambda < \mu$.

where $H_p(x) := x \ln\left(\frac{x}{p}\right) + (1 - x) \ln\left(\frac{1-x}{1-p}\right)$ is an entropy function.

Proof. Notice that the two statements are entirely symmetric, so we will prove just the first one. Define $m := \mu + \lambda$; we want $\mathbb{P}[X \geq m]$. The first step is to introduce a variable t (by way of the *Laplace* transform) and use Markov's to get

$$\begin{aligned}
\mathbb{P}[X \geq m] &= \mathbb{P}[e^{tX} \geq e^{tm}] && \text{(for any } t > 0\text{)} \\
&\leq e^{-tm}\mathbb{E}[e^{tX}] && \text{(by Markov)} \\
&= e^{-tm} \prod_{i=1}^n \mathbb{E}\left[e^{tX_i}\right] && \text{(by independence)} \\
&= e^{-tm} \prod_{i=1}^n \left(p_i e^t + 1 - p_i\right) && \text{(by distributions of the } X_i\text{)} \\
&\leq e^{-tm}(e^t p + 1 - p)^n && \text{(by AM/GM for concavity of } \ln((e^t - 1)z + 1)\text{)} \\
&= \exp\left\{n \ln(pe^t + (1 - p)) - tm\right\}
\end{aligned}$$

Now we use calculus to find that the optimum is at $e^t = \frac{m(1-p)}{(n-m)p} > 1$, so plugging it in we find

$$\begin{aligned}
\mathbb{P}[X \geq m] &\leq \exp\left\{n \ln\left(\frac{m(1-p)}{n-m} + 1 - p\right) - m \ln\left(\frac{m(1-p)}{(n-m)p}\right)\right\} \\
&\vdots \\
&= \exp\left(-nH_p(p + \frac{\lambda}{n}\right)
\end{aligned}$$

after we tidy up our expression and use $m = np + \lambda$. □

No one actually goes around carrying these inequalities in their heads, but the point of going through the proof is to use these general techniques when we're presented with any sum of indicator variables.

Corollary 35. We have that

$$\left.\begin{matrix}\mathbb{P}[X \leq \mu - \lambda] \\ \mathbb{P}[X \geq \mu + \lambda]\end{matrix}\right\} \leq \exp\left(-\frac{2\lambda^2}{n}\right).$$

Additionally, if $X_i \in [a_i, b_i]$, we have the refinement

$$\mathbb{P}[|X - \mu| \geq \lambda] \leq \exp\left(-\frac{2\lambda^2}{\sum (b_i - a_i)^2}\right).$$

Corollary 36 (Angluin/Valiant)**.** We have the (slightly) asymmetric bounds

$$\mathbb{P}[X \leq (1-\beta)\mu] \leq \exp\left[-\mu(\beta + (1-\beta)\ln(1-\beta))\right] \leq \exp(-\frac{beta^2\mu}{2})$$

and

$$\mathbb{P}[X \geq (1+\beta)\mu] \leq \exp\left[-\mu(-\beta + (1+\beta)\ln(1+\beta))\right] \leq \begin{cases}\exp(-\frac{\beta^2\mu}{2+\beta}) & \beta > 0 \\ \exp(-\frac{\beta^2\mu}{3}) & 0 < \beta < 1\end{cases}$$

Proof. Plug in $\lambda = \beta\mu$ in the above theorem and use $\ln(1+x) < x$. You can further clean them up using the first few terms of the Taylor Expansion. □

You should actually keep this bound around in your head.

Example 13.1 (Fair Coin Flips)**.** Suppose we flip n fair coins, so that $\mu = n/2$. Then

$$\mathbb{P}[|X - \mu| \geq \lambda] \leq 2e^{-2\lambda^2/n}$$

by Corollary 1.

Example 13.2 (Biased Coin Flips)**.** Now we flip n biased coins with $\mathbb{P}[X_i = 1] = 3/4$, so that $\mu = 3n/4$. Then

$$\begin{aligned}\mathbb{P}[X \leq n/2] &= \mathbb{P}[X \leq (1 - 1/3)\mu] \\ &\leq \exp\left(-\frac{(1/3)^2}{2} \cdot \frac{3n}{4}\right) \\ &= \exp\left(-\frac{n}{24}\right)\end{aligned}$$

by Corollary 2.

13.2 Randomized Routing

This problem is due to Valiant/Brebner.

In this set up, we have an n-dimensional hypercube $\{0,1\}^n$ which has $N = 2^n$ vertices. At each vertex, there is a packet i which has a destination vertex $\sigma(i)$ it wishes to route to (essentially these packets are a permutation), but at any given time step, only one packet can traverse a single edge in one direction (but multiple can do so simultaneously, and in different directions). If there are multiple packets that wish to traverse the same direction on an edge, then one packet goes and the others wait (say using a queueing system) for the next time step. We want to ask what the maximum amount of delay we can guarantee first.

There is a negative result in the case of deterministic strategies first.

Theorem 37. For any deterministic oblivious routing strategy, there exists a permutation σ that requires $\Omega(\sqrt{2^n/n})$ steps.

But using randomization, we can achieve the following result.

Theorem 38. There exists a randomized oblivious scheme that terminates in $O(n)$ steps w.h.p.

To achieve this, we can use the following rather trivial 2-phase scheme:

1. Each packet i chooses an intermedaite destination $\tau(i)$ u.a.r. and goes there using bit-fixing path (τ is not necessarily a permutation).
2. Each packet goes from $\tau(i) \to \sigma(i)$ using a bit fixing path.

Proof. Let $D(i)$ be the delay suffered by packet i in Phase 1. Phase 1 and 2 are equivalent, so we will focus on the former. Using the Chernoff Bound, we can achieve the following result.

Claim. $\mathbb{P}[D(i) > \frac{7}{2}n] \le e^{-2n}$.

Corollary 39.

$$\mathbb{P}[\text{any packet takes} > 9n/2 \text{ steps in Phase 1}] \le 2^n e^{-2n} < 2^{-n},$$

so overall,

$$\mathbb{P}[\text{packets take} > 9n \text{ steps overall}] < 2^{-(n+1)}.$$

Claim (Claim'). $D(i) \le |S(i)|$, where $S(i) := \{j : \text{ path of packet } j \text{ intersects path of packet } i\}$.

This is not a trivial statement, as any packet j could intersect multiple times with i's path and cause more delay.

Claim (Claim"). For all i, $\mathbb{P}[|S_i| > \frac{7}{2}n] \le e^{-2n}$.

Proof of Claim". Fix i. Let $H_{ij} = 1$ if $P_i \cap P_j \neq \emptyset$ and 0 otherwise. Then $|S_i| = \sum_{j \neq i} H_{ij}$. Calculating each of the H_{ij}'s is actually quite messy, as they are not independent, so instead we will look at their average.

We claim that $\mathbb{E}[\sum_{j \neq i} H_{ij}] \le n/2$. Let $T(e_t) =$ the number of paths that pass through edge e_t. Then

$$\mathbb{E}[T(e_t)] = \frac{\mathbb{E}[\text{total length of all paths}]}{\#\text{ total edges}} = \frac{N \times n/2}{Nn} = \frac{1}{2}.$$

So

$$\mathbb{E}\left[\sum_{j \neq i} H_{ij}\right] \le \sum_{i=1}^{k} \mathbb{E}[T(e_i)] = k/2 \le n/2.$$

Now use Angluin-Valiant with $\mu = n/2$, $\beta = 6$ to achieve our Chernoff bound on $7n/2$ steps. □

□

14 Thursday, March 5th