Problem Set 2 CS265, Autumn 2023

Due: 10/13 (Friday) at 11:59pm on Gradescope

Please follow the homework policies on the course website.

---

1. **(8 pt.) [Counting small cuts.]**

   Recall that a cut of an undirected graph $G = (V, E)$ is a partition of the vertices V into nonempty disjoint sets A and B. A *min cut* of G is a cut that minimizes the number of edges that cross the cut (have one endpoint in A and one in B).

   In the following problems, assume G is a connected graph on n vertices (i.e., there is no cut with 0 edges that cross it).

   (a) **(2 pt.)** A graph may have many possible min cuts. Prove that G has *at most* $n(n-1)/2$ min cuts.

   (b) **(2 pt.)** Show that part (a) is tight; for every $n \geq 2$, give a connected graph on n vertices with exactly $n(n-1)/2$ min cuts.

   (c) **(4 pt.)** Let α be a positive integer. Suppose that any min cut of G has k edges that cross the cut. An α-*small* cut of G is a cut that has at most αk edges that cross the cut. Prove that the number of such cuts is at most $O(n^{2\alpha})$.

   [**Note:** If you find it easier, you'll still get full credit if you prove a bound of $O((2n)^{2\alpha})$.]

   [**HINT:** *Consider stopping Karger's algorithm early and then outputting a random cut in the contracted graph. What is the probability that this returns a fixed α-small cut of G?*]

   (d) **(0 pt.) [Optional: this won't be graded]** Let $f(n, \alpha)$ be the maximum number of α-small cuts that an n vertex graph can have. What are the tightest upper and lower bounds you can find for $f(n, \alpha)$?

SOLUTION:

(a) Fix a min cut of G. We know from class that when Karger's algorithm is run on G, outputs this particular min cut with probability at least $2/(n(n-1))$. Suppose that G has t min cuts $C_1, C_2, ..., C_t$. Applying the above argument for every min cut, we have that

$$1 \geq \sum_{i=1}^{t} \Pr[\text{Karger's outputs } C_i] \geq t \cdot \frac{2}{n(n-1)}$$

and so $t \leq n(n-1)/2$.

(b) Consider a cycle on n vertices. Let's consider the minimum possible cut. If there were only one edge that goes across a given cut, then the endpoints of every other edge are on the same side of the cut. But all the other edges form a path of length $n-1$ including all the vertices, so then every vertex must be on the same side of the cut. This is a contradiction.

The cut that separates any vertex from the rest has two edges that go across the cut, so every min cut must have size 2. Moreover, for every pair of edges, there is a cut where these two edges are the only ones that cross the cut. Hence, there are $\binom{n}{2} = n(n-1)/2$ min cuts in this graph.

(c) Fix an α-small cut C. Suppose that we run Karger's algorithm until 2α vertices remain, and then we output a uniformly random cut of the contracted graph. We consider the probability that this outputs our fixed cut.

First, we consider the probability that after the steps of Karger's algorithm, the α-small cut remains. Following a similar line of attack to the proof correctness for Karger's algorithm, we define E_i denote the event that we do not contract an edge crossing C in the ith step of the algorithm. We have

$$\Pr[C \text{ remains after contractions}] = \Pr[E_1]\Pr[E_2|E_1]\cdots\Pr[E_{n-2\alpha} \mid E_1, ..., E_{n-2\alpha-1}].$$

While the all of the min cuts may be destroyed by the contractions, we still know that the size of the min cut can only increase. Therefore, in the ith step of the algorithm, there still are at least $nk/2$ edges remaining. Since the number of edges that cross C is at most αk, we have

$$\Pr[E_i \mid E_1, ..., E_{i-1}] \geq 1 - \frac{\alpha k}{(n-i+1)k/2} = 1 - \frac{2\alpha}{n-i+1} = \frac{n-i+1-2\alpha}{n-i+1}.$$

It follows that,

$$\begin{aligned}\Pr[C \text{ remains after contractions}] &\geq \frac{n-2\alpha}{n} \cdot \frac{n-2\alpha-1}{n-1} \cdot \dots \cdot \frac{1}{2\alpha+1} \\ &= \frac{1 \cdot 2 \cdot \dots \cdot 2\alpha}{n(n-1)\cdots(n-2\alpha+1)} \\ &\geq \frac{2^{2\alpha-1}}{n^{2\alpha}}.\end{aligned}$$

Finally, note that in a t vertex graph, there are $2^{t-1}-1$ cuts, since each vertex has two sides of the cut to choose from, but this double counts by a factor of 2 and includes the non-cut where every vertex is on one side.

Since 2α vertices remain in the end, the probability that we choose any particular cut is greater than $1/2^{2\alpha-1}$. In particular, the probability that we output C, given that C remains after the contractions is at least $1/2^{2\alpha-1}$. This gives us

$$\Pr[\text{output } C] \geq \frac{1}{2^{2\alpha-1}} \cdot \frac{2^{2\alpha-1}}{n^{2\alpha}} = \frac{1}{n^{\alpha}}.$$

By the same argument in part (a), it follows that there are at most $n^{2\alpha}$ α-small cuts.

(d) We can show the following bounds

$$\binom{n}{2\alpha} \leq f(n,\alpha) \leq \sqrt{\frac{\pi\alpha}{2}}\binom{n}{2\alpha}$$

which pins down $f(n,\alpha)$ within a $O(\sqrt{\alpha})$ factor. For the lower bound, it's not hard to see that once again, for the cycle on n vertices, every set of 2α edges are the edges that go across some cut (In fact, this means that $f(n,\alpha) \geq \sum_{k=1}^{\alpha} \binom{n}{2k}$, which can give an improvement for large α).

For the upper bound, the key observation is that we should actually stop Karger's earlier, since the probability guarantee degrades by more than a factor of $\frac{1}{2}$ once the number of vertices goes below 4α. Stopping at this point, we get

$$\begin{aligned}
\Pr[C \text{ remains after contractions}] &\geq \frac{n-2\alpha}{n} \cdot \frac{n-2\alpha-1}{n-1} \cdot \dots \cdot \frac{2\alpha+1}{4\alpha+1} \\
&= \frac{(2\alpha+1)\cdot(2\alpha+2)\cdot\dots\cdot 4\alpha}{n(n-1)\cdots(n-2\alpha+1)} \\
&= \frac{\binom{4\alpha}{2\alpha}}{\binom{n}{2\alpha}} \\
&\geq \frac{\frac{1}{\sqrt{2\pi\alpha}}2^{4\alpha}}{\binom{n}{2\alpha}}
\end{aligned}$$

using Stirling's approximation. Now 4α vertices remain in the end, so there are at most $2^{4\alpha-1}$ possible cuts. We conclude that

$$\Pr[\text{output } C] \geq \frac{1}{2^{4\alpha-1}} \cdot \frac{\frac{1}{\sqrt{2\pi\alpha}}2^{4\alpha}}{\binom{n}{2\alpha}} = \frac{2}{\sqrt{2\pi\alpha}\binom{n}{2\alpha}}$$

which gives us the desired result.

2. **(12 pt.) [Tightness of Markov's and Chebyshev's Inequalities]**

   (a) **(4 pt.)** Show that Markov's inequality is tight. Specifically, for each value $c > 1$, describe a distribution D_c supported on non-negative real numbers such that if the random variable X is drawn according to D_c then (1) $\mathbb{E}[X] > 0$ and (2) $\Pr[X \geq c\mathbb{E}[X]] = 1/c$.

   (b) **(4 pt.)** Show that Chebyshev's inequality is tight. Specifically, for each value $c > 1$, describe a distribution D_c supported on real numbers such that if the random variable X is drawn according to D_c then (1) $\mathbb{E}[X] = 0$ and $\mathrm{Var}[X] = 1$ and (2) $\Pr[|X - \mathbb{E}[X]| \geq c\sqrt{\mathrm{Var}[X]}] = 1/c^2$.

   (c) **(4 pt.)** [One-sided version of Chebyshev's Inequality] Prove a one-sided bound on the distribution of a random variable X given its variance. That is, if $\mathrm{Var}[X] = 1$, what the best upper bound on $\Pr[X - \mathbb{E}[X] \geq t]$? Give your answer in terms of t. Prove your bound (a) is true and (b) is tight by coming up with a variable X with distribution D_t and variance 1 for which $\Pr[X - \mathbb{E}[X] \geq t]$ equals your answer.

SOLUTION:

(a) Let D_c be the distribution where $X = c$ with probability $1/c$ and $X = 0$ with probability $1 - 1/c$, such that $\mathbb{E}[X] = 1$.

(b) Let D_c be the distribution where $X = c$ with probability $\frac{1}{2c^2}$, $X = -c$ with probability $\frac{1}{2c^2}$, and $X = 0$ with probility $1 - 1/c^2$, such that $\mathbb{E}[X] = 0$ and $\mathrm{Var}[X] = 1$.

(c) The correct bound is $\Pr[X - \mathbb{E}[X] \geq t] \leq \frac{1}{1+t^2}$. To prove this, for any y,

$$\begin{aligned} \Pr[X - \mathbb{E}[X] \geq t] &= \Pr[X - \mathbb{E}[X] + y \geq t + y] && (1)\\ &\leq \Pr[(X - \mathbb{E}[X] + y)^2 \geq (t + y)^2] && (2)\\ &\leq \frac{\mathbb{E}[(X - \mathbb{E}[X] + y)^2)]}{(t+y)^2} && (3)\\ &= \frac{\mathrm{Var}[X] + y^2}{(t+y)^2} && (4)\\ &= \frac{1 + y^2}{(t+y)^2}. && (5) \end{aligned}$$

To get the tightest bound, choose y to minimize $\frac{1+y^2}{(t+y)^2}$. We take the derivative to do this:

$$\frac{d}{dy}\frac{1+y^2}{(t+y)^2} = \frac{(t+y)^2(2y) - (1-y^2)(2(t+y))}{(t+y)^4}, \tag{6}$$

At the minimum, the derivative must be zero, so we have $(t+y)^2(2y) - (1+y^2)(2(t+y)) = 0$, so $(t+y)y = (1+y^2)$, and thus $y = 1/t$. Plugging this in, we get the bound

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \frac{1 + 1/t^2}{(t + 1/t)^2} = \frac{1+t^2}{(1+t^2)^2} = \frac{1}{1+t^2}. \tag{7}$$

To show tightness, consider the distribution D_t which equals t with probability $\frac{1}{1+t^2}$ and equals $-1/t$ with probabilty $\frac{t^2}{1+t^2}$. It is easy to check that $\mathbb{E}[X] = 0$ and $\mathrm{Var}[X] = 1$.

3. **(9 pt.) [Cutting Losses and Starting Fresh]** Suppose someone gives you a device with a button that, when pressed, runs a randomized algorithm for problem X the with the following guarantees: 1) The algorithm has expected runtime 1 minute, and 2) when the algorithm terminates, it always returns a correct answer. If you press the button before the algorithm terminates, the device simply resets and starts running the same algorithm again (with new/independent randomness).

   (a) **(3 pt.)** Suppose I have 6 minutes to solve the problem—after 6 minutes even a correct answer is useless to me. How could I use the device to answer the problem within 6 minutes with a probability of at least $1 - 1/3^2$? [Hint: If I push the button just once, by Markov's inequality, the probability I don't get my answer within 6 minutes might be as large as 1/6. After pushing the button, how long should I wait until I push the button again?]

(b) **(6 pt.)** Can you come up with a protocol for re-pushing the button does better than $1-1/3^2$? If so, describe one such strategy and prove that its success probability exceeds $1-1/3^2$ by at least 0.001. If not, prove that there is a distribution over runtimes such that it is impossible to improve upon this success probability. [Hint: If Markov's inequality is tight, what does that tell you about the distribution of the runtimes, and can you exploit that?]

(c) **(0 pt.)** What is an optimal protocol, and what is the best probability of success that you can provably always get (no matter the runtime distribution, given that its expectation is 1)? Feel free to answer this either in the case of 6 minutes, or in the limit as the total time gets large.

SOLUTION:

(a) Restart after 3 seconds. By Markov's inequality, the probability the algorithm runs past 3 seconds is at most 1/3, and we get 2 3-second runs in the 6 second time budget, so the probability we don't get a solution is the probability that both the first and second run take more than 3 seconds, namely $1/3^2$.

(b) There are many valid solutions. The high level insight is that if Markov's inequality is close to being tight, then there must be a very good probability that the runtime is very small (close to 0) to offset the significant probability of being large. Hence if we do a short run, that has a reasonable success probability if our longer runs have a close-to-worst-case success probability.

Here is one concrete instantiation of this approach. Let X denote the random variable representing the runtime. X is non-nonnegative and has expectation 1, and hence if $Pr[X > 2.9] > 1/3 - 0.02$, then in order for the expectation to be 1, $Pr[X < 0.2]$ needs to be fairly large. Namely assuming $Pr[X > 2.9] > 1/3 - 0.02$, then $1 = E[X] \geq 2.9(1/3 - 0.02) + 0.2 \cdot Pr[X \in [0.2, 2.9]]$, which implies that $\Pr[X \in [0.2, 2.9]] \leq (1 - 2.9(1/3 - 0.02)/0.2 < 0.46$. Hence $\Pr[X \leq 0.2] \geq 1 - 1/3 - 0.46 > 0.21$.. So, consider running the algorithm for 2.9 minutes, then restart, run for another 2.9 minutes, and then for another 0.2 minutes. If $Pr[X > 2.9] \leq 1/3 - 0.02$, then the probability that the first two runs both fail is already at most $(1/3-0.02)^2 < 0.1$ If $Pr[X > 2.9] > 1/3-0.02$, then the probability the final run (the 0.2 minute run) is successful is at least 0.21, and since, by Markov's inequality, $Pr[X > 2.9] < 1/2.9$, the probability all 3 runs fail is at most $(1/2.9^2)(1 - 0.21) < 0.1$. Hence we've given a proof that the probability of failure is at most 0.1, which is better than the $1/3^2 \approx 0.11$ probability of failure of part (a).

(c) In general we won't post solutions to bonus parts, but feel free to discuss with us in office hour.

4. **(0 pt.) [This whole problem is optional and will not be graded.]** In this problem, you'll analyze a different primality test than we saw in class. This one is called the *Agrawal-Biswas Primality test.*

Given a degree d polynomial $p(x)$ with integer coefficients, for any polynomial $q(x)$ with integer coefficients, we say $q(x) \equiv t(x) \mod (p(x), n)$ if there exists some polynomial $s(x)$ such that $q(x) = s(x) \cdot p(x) + t(x) \mod n$. (Here, we say that $\sum_i c_i x^i = \sum_i c'_i x^i \mod n$ if and only if $c_i = c'_i \mod n$ for all i.) For example, $x^5 + 6x^4 + 3x + 1 \equiv 3x + 1 \mod (x^2 + x, 5)$, since $(x^3)(x^2 + x) + (3x + 1) = x^5 + x^4 + 3x + 1 \equiv x^5 + 6x^4 + 3x + 1 \mod 5$.

Agrawal-Biswas Primality Test.
Given n:

- If n is divisible by 2,3,5,7,11, or 13, or is a perfect power (i.e. $n = c^r$ for integers c and r) then output **composite**.
- Set d to be the smallest integer greater than $\log n$, and choose a random degree d polynomial with leading coefficient 1:

$$r(x) = x^d + c_{d-1}x^{d-1} + \ldots + c_1 x + c_0,$$

  by choosing each coefficient c_i uniformly at random from $\{0, 1, \ldots, n-1\}$.
- If $(x + 1)^n \equiv x^n + 1 \mod (r(x), n)$ then output **prime**, else output **composite**.

Consider the following theorem (you can assume this if you like, or for even more optional work, try to prove it!):

Theorem 1 (Polynomial version of Fermat's little theorem)**.**

- *If n is prime, then for any integer a, $(x - a)^n = x^n - a \mod n$.*
- *If n is not prime and is not a power of a prime, then for any a s.t. $gcd(a, n) = 1$ and any prime factor p of n, $(x - a)^n \neq x^n - a \mod p$.*

First, show that if n is prime, then the Agrawal-Biswas primality test will always return **prime**.

Now, we will prove that if n is composite, the probability over random choices of $r(x)$ that the algorithm successfully finds a witness to the compositeness of n (and hence returns **composite**) is at least $\frac{1}{4d}$.

(a) Using the polynomial version of Fermat's Little Theorem, and the fact that, for prime q, every polynomial over $\mathbb{Z}_q$ that has leading coefficient 1 (i.e. that is "monic") has a unique factorization into irreducible monic polynomials, prove that the number of irreducible degree d factors that the polynomial $(x + 1)^n - (x^n + 1)$ has over $\mathbb{Z}_p$ is at most n/d, where p is any prime factor of n. (A polynomial is irreducible if it cannot be factored, for example $x^2 + 1 = (x + 1)(x + 1) \mod 2$ is not irreducible over $\mathbb{Z}_2$, but $x^2 + 1$ is irreducible over $\mathbb{Z}_3$.)

[**HINT:** *Even though this question sounds complicated, the proof is just one line...*]

(b) Let $f(d, p)$ denote the number of irreducible monic degree d polynomials over $\mathbb{Z}_p$. Prove that if n is composite, and not a power of a prime, the probability that $r(x)$ is a witness to the compositeness of n is at least $\frac{f(d,p) - n/d}{p^d}$, where p is a prime factor of n.

[**HINT:** *p^d is the total number of monic degree d polynomials over $\mathbb{Z}_p$.*]

(c) Now complete the proof, and prove that the algorithm succeeds with probability at least $1/(4d)$, leveraging the fact that the number of irreducible monic polynomials of degree d over $\mathbb{Z}_p$ is at least $p^d/d - p^{d/2}$. (You should be able to prove a much better bound, though $1/4d$ is fine.)
[**HINT:** *You will also need to leverage the fact that we chose $d > \log n$ and also explicitly made sure that n has no prime factors less than* 17.]

SOLUTION:

First, by the polynomial version of FLT (with $a = -1$), we know that $(x^n + 1) - (x^n + 1) \equiv 0 \mod n$ when n is prime, which means that this is true mod any polynomial as well. This means that when n is prime, the algorithm always outputs "prime."

(a) Let c be the leading coefficient of $(x^n+1)-(x^n+1)$ over $\mathbb{Z}_p$. Since p is a prime and c (by definition) is nonzero, c has an inverse c^{-1}. Then the polynomial $c^{-1}((x^n+1)-(x^n+1))$ is monic, and has degree at most n. Writing this polynomial as its factorization into irreducible polynomials, we can see that the number of irreducible degree factors must be at most n/d (else the polynomial would have degree $> n$).

(b) Suppose that n is composite and not a power of a prime. First we prove a couple of results that will help us trade between mod n and mod p.

For each coefficient c_i of $r(x)$, c_i is equally likely to be anything mod p since p divides n (the elements of $\mathbb{Z}_n$ that map to $t \in \mathbb{Z}_p$ are always the n/p elements $t + kp$ for $0 \le k < n/p$). Hence, if we interpret $r(x)$ modulo p, then each of the p^d monic degree d polynomials is equally likely to be generated.

In particular, the probability that $r(x)$ mod p is one of the monic irreducible degree d polynomials that is *not* a factor of $(x+1)^n - (x^n+1)$ is at least $\frac{f(d,p)-n/d}{p^d}$, since each of the p^d polynomials is equally likely, there are $f(d,p)$ polynomials that are monic and irreducible, and at most n/d of them are factors of $(x+1)^n - (x^n+1)$ (by part a).

Now we claim that

$$(x+1)^n - (x^n+1) \equiv 0 \pmod{(r(x), n)} \implies (x+1)^n - (x^n+1) \equiv 0 \pmod{(r(x), p)}.$$

Suppose that the left side is true, and let $(x+1)^n - (x^n+1) \equiv s(x)r(x) + t(x) \pmod{(r(x), p)}$. Since $\mathbb{Z}_p$ is a subgroup of $\mathbb{Z}_n$, it follows that $(x+1)^n - (x^n+1) \equiv s(x)r(x) + t(x) \pmod{(r(x), n)}$, which means that $t(x) \equiv 0 \pmod{n}$. This means that each coefficient in $t(x)$ is divisible by n, which means that they are also divisible by p, and so $t(x) \equiv 0 \pmod{p}$ as well. Hence, $(x+1)^n - (x^n+1) \equiv 0 \pmod{(r(x), p)}$.

The contrapositive of this tells us that

$$(x+1)^n - (x^n+1) \not\equiv 0 \pmod{(r(x), p)} \implies (x+1)^n - (x^n+1) \not\equiv 0 \pmod{(r(x), n)}.$$

Hence, if $r(x)$ mod p is one of the monic irreducible degree d polynomials that is *not* a factor of $(x+1)^n - (x^n+1)$, then we know that the left statement is true, and therefore the right statement is also true. This is exactly the condition the algorithm uses to say that n is composite. Hence, if $r(x)$ mod p is one of the monic irreducible degree

d polynomials that is not a factor of $(x+1)^n - (x^n + 1)$, then the algorithm correctly outputs composite, and since this happens with probability at least $\frac{f(d,p)-n/d}{p^d}$, this is a lower bound on the success probability.

(c) Our probability of success is at least

$$\frac{f(d,p) - n/d}{p^d} \geq \frac{p^d/d - p^{d/2} - n/d}{p^d} = \frac{1}{d} - \frac{1}{p^{d/2}} - \frac{n}{dp^d}.$$

Since $d > \log_2 n$, we have $n < 2^d$. Hence

$$\frac{n}{dp^d} < \frac{1}{d} \cdot \left(\frac{2}{p}\right)^d \leq \frac{1}{d} \cdot \left(\frac{2}{17}\right)^d \leq \frac{2}{17d}.$$

We also have for all $d \geq 1$,

$$4d \leq 4^d < 17^{d/2} \leq p^{d/2}$$

which means that $\frac{1}{p^{d/2}} < 1/4d$. Hence, the probability of success is at least

$$\frac{1}{d} - \frac{1}{4d} - \frac{2}{17d} \geq \frac{1}{4d}$$

as desired.