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 $f(n, \alpha)$?

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.

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.

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.*]