Advanced Algorithms (3501)

Spring Semester, 2023

Problem Set 1

Published on: 01/05/2023
Submit by: 16/05/2023, 23:59

1. In this exercise we'll prove that the existence of a zero error randomized algorithm (whose expected running time is polynomial) implies the existence of a polynomial time bounded error randomized algorithm (whose running time is polynomial in the worst case).

   Let ZPP be the set of languages for which there's a zero-error randomized algorithm that decides them, whose running time is polynomial *in expectation*.

   Let $\mathsf{RP} = \mathsf{BPP}(0, 1/2)$ and $\mathsf{coRP} = \mathsf{BPP}(1/2, 1)$.

   (a) Show that one can equivalently define ZPP as the class of languages A for which there's a randomized algorithm M with the following properties:

      i. The running time of M is polynomial *in the worst case* on every input x.

      ii. Given an input x, M either answers whether x is in A or says "I don't know".

      iii. If M doesn't say "I don't know", its answer is always correct.

      iv. M says "I don't know" with probability as most $1/3$ (the probability is over the random coins of M and not over the inputs).

   (b) Show that $\mathsf{RP} \subseteq \mathsf{BPP}$ and similarly $\mathsf{coRP} \subseteq \mathsf{BPP}$.

   (c) Show that $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$.

2. Prove the *Schwartz Zippel Lemma*: Let $f(x_1, \ldots, x_n)$ be a nonzero degree-d polynomial over $\mathbb{Q}$. Let $S \subseteq \mathbb{Q}$ be a finite set. Then

$$\Pr_{\alpha_1,\ldots,\alpha_n \in S}[f(\alpha_1, \ldots, \alpha_n) = 0] \leq \frac{d}{|S|},$$

   where $\alpha_1, \ldots, \alpha_n$ are picked uniformly at random and independently from S.

   (Hint: use induction on n. For the base case, you may use (without proof) the fact that a non-zero degree-d polynomial has at most d roots. For the induction step, write f as $\sum_{i=0}^{d} x_n^i f_i(x_1, \ldots, x_{n-1})$ for some polynomials $f_0, \ldots, f_d$ on $n-1$ variables, and proceed from there).

3. Let G be a d-regular graph and let A be its normalized adjacency matrix. As we defined in class, let $B = \frac{1}{2}I + \frac{1}{2}A$. Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of B in descending order so that

$$0 \leq \lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_2 \leq \lambda_1 = 1.$$

In this exercise we'll prove the following claim that we've stated in class: if G is connected, then $\lambda_2 \leq 1 - \frac{1}{4dn^3}$.

(a) Let x be a normalized eigenvector corresponding to λ_2 such that $\|x\|_2 = 1$. Show that there's a coordinate i such that $|x_i| \geq \frac{1}{\sqrt{n}}$.

(b) Suppose without loss of generality that for the coordinate i from the previous item, $x_i \geq \frac{1}{\sqrt{n}}$. Show that there's a different coordinate j such that $x_j \leq 0$. (Hint: use the fact that $\mathbf{1}$ is an eigenvector corresponding to λ_1, and that eigenvectors corresponding to different eigenvalues are orthogonal).

(c) Show that there's an edge (u, v) in G such that $|x_u - x_v| \geq \frac{1}{n\sqrt{n}}$. (Hint: use the assumption that G is connected and follow a path from x_i to x_j.)

(d) Justify the following set of equalities, and conclude the upper bound on λ_2.

$$\begin{aligned}
\lambda_2 = \lambda_2 \cdot \langle x, x \rangle = \langle \lambda_2 x, x \rangle = \langle Bx, x \rangle &= \sum_{k,\ell} B_{k,\ell} \cdot x_k x_\ell \\
&= \frac{1}{2} \sum_{k,\ell} B_{k,\ell} \left[x_k^2 + x_\ell^2 - (x_k - x_\ell)^2 \right] \\
&= \sum_k x_k^2 - \frac{1}{2} \sum_{k,\ell} B_{k,\ell} \cdot (x_k - x_\ell)^2 = 1 - \frac{1}{2} \sum_{k,\ell} B_{k,\ell} \cdot (x_k - x_\ell)^2
\end{aligned}$$

4. In the *equality problem*, each of two players, Alice and Bob, gets as an input an n-bit string. Each player sees their own input but can't see the other player's input. Alice and Bob would like to find out whether their inputs are equal or not. It can be shown that if Alice and Bob are deterministic then this task requires exchanging at least n bits (for example, Alice can send Bob her entire string). Devise a *randomized* protocol that enables them to find out whether their inputs are equal or not, and only requires sending $O(\log n)$ bits in the worst case. The error probability of the protocol should be at most $1/3$ (that is, the behavior of Alice and Bob may be random, and for every pair of inputs, the players err with probability at most $1/3$).

(Hint: one possible way to do this is to interpret the inputs as binary integers of size at most $N = 2^n$ and to compute their remainder modulo p for a random prime p picked from a certain suitable interval. You may use properties of prime numbers we mentioned in class. Think: how many distinct prime divisors can a number of size at most N have?)