CS6845: Pseudorandomness 15 Jan 2018

1. Introduction

1.1 Introduction

Randomization is used extensively in computer science. It is used to design efficient algorithms that almost always give the right answers, and also in proving the existence of combinatorial objects. We will see examples of both in this lecture.

1.2 Randomization in algorithms

Consider the following problem:

Problem 1.1. *Given a polynomial $f(x_1, x_2, \ldots, x_n) \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, test if f is the zero polynomial?*

Here what we are interested in is knowing whether f is the zero polynomial when written down explicitly as a sum of monomials, and not if f evaluates to 0 on all inputs to $x_1, x_2, \ldots, x_n$. The two notions are different if the degree of polynomial is at least as large as the size of the field. For instance, for the polynomial $f(x) = x^2 - x \in \mathbb{F}_2[x]$, the answer to Problem 1.1 is "no" whereas f evaluates to zero at each point of $\mathbb{F}_2$.

But, how is the polynomial "given"?

- If f is given as a list of coefficients of the monomials, then the problem is easy to solve. Just check if all the coefficients are zero.
- What if the polynomial is given as a "black-box": you are allowed to give an input $\alpha_1, \alpha_2, \ldots, \alpha_n$, and the black-box returns $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Lemma 1.2 (DeMillo-Lipton-Schwartz-Zippel lemma)**.** *Let f be a n-variate non-zero polynomial over $\mathbb{F}$ of degree d, and let $S \subseteq \mathbb{F}$. Then,*

$$\Pr_{\alpha_1, \alpha_2, \ldots, \alpha_n \in_r S}[f(\alpha_1, \alpha_2, \ldots, \alpha_n) = 0] \leq \frac{d}{|S|}.$$

Use the lemma to obtain a randomized algorithm. What are its guarantees? What happens if the polynomial has degree as large as the field?

1.3 Probabilistic method

Let $G(V, E)$ be a graph on n vertices and m edges. A *cut* of the graph is a partition of $V = V_1 \cup V_2$. The *size* of the cut is the number of edges that have one end point in V_1 and the other in V_2.

Problem 1.3 (MAX-CUT)**.** *Given a graph $G(V, E)$ find the cut with the maximum size.*

This problem is known to be NP-hard. In the next lecture we will see how to use the probabilistic method to show that a cut with a large size always exists.

Lemma 1.4. *A graph $G(V, E)$ on n vertices with m edges has a cut of size at least $m/2$.*