Algebraic Techniques in TCS (Fall'21)	Assigned: September 15, 2021
Homework 1	
Instructor: *Josh Alman*	Due: September 28, 2021 at 3:45 pm

Collaboration on homework is encouraged. However, you must write up solutions by yourself and understand everything that you hand in. For each question on the problem set, please write a list of everyone you collaborated on that problem with. You may not seek out answers from other sources without permission.

On all the problems, you may assume we have an algorithm which runs in time $O(n^{2.373})$ for multiplying two $n \times n$ matrices whose entries are all integers between $-n^{100}$ and n^{100}. You must prove that all your algorithms have the required running times and properties.

1 Polynomials and Linear Algebra

(a) Prove that for any Boolean function $f : \{0,1\}^n \to \{0,1\}$, there is exactly one multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that $p(x) = f(x)$ for all $x \in \{0,1\}^n$. (Recall that a multilinear polynomial p in n variables can be written as $p(x_1, \ldots, x_n) = \sum_{S \subseteq \{1,2,\ldots,n\}} a_S \prod_{i \in S} x_i$ for coefficients $a_S \in \mathbb{R}$.)

(b) For each of the following functions $f : \{0,1\}^n \to \{0,1\}$, find (with proof) the multilinear polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that $p(x) = f(x)$ for all $x \in \{0,1\}^n$. (In other words, give a formula/expression for the coefficient a_S of $\prod_{i \in S} x_i$ in p for each $S \subseteq \{1, 2, \ldots, n\}$.)

- $f(x) = OR(x) = \begin{cases} 0 & \text{if } x_1 = \cdots = x_n = 0, \\ 1 & \text{otherwise.} \end{cases}$
- $f(x) = PARITY(x) = \begin{cases} 0 & \text{if } \sum_{i=1}^n x_i \text{ is even,} \\ 1 & \text{if } \sum_{i=1}^n x_i \text{ is odd.} \end{cases}$

2 Approximating the AND Function

In this problem, we will design a polynomial $T_m : \mathbb{R} \to \mathbb{R}$ such that $-1 \leq T_m(x) \leq 1$ for all $-1 \leq x \leq 1$, but $T_m(x)$ grows quickly for $x > 1$. We will then use it to design a polynomial which approximates the AND function.

(a) Prove that for each integer $m \geq 0$, there is a polynomial $T_m : \mathbb{R} \to \mathbb{R}$ of degree m such that $T_m(\cos x) = \cos(m \cdot x)$ for all $x \in \mathbb{R}$. Hint: use the trigonometric identity $\cos((m+1)x) = 2\cos(x)\cos(mx) - \cos((m-1)x)$ to define T_m recursively.

(b) Prove that if $x \geq 1$, then

$$T_m(x) = \frac{1}{2}\left(\left(x + \sqrt{x^2 - 1}\right)^m + \left(x - \sqrt{x^2 - 1}\right)^m\right).$$

(c) Prove that if $0 < \varepsilon < 1/4$, then $T_m(1+\varepsilon) \geq \frac{1}{2}e^{m\sqrt{\varepsilon}}$. Hint: use the fact that for such ε, we have $1 + \sqrt{2\varepsilon} \geq e^{\sqrt{\varepsilon}}$.

(d) Prove that for any integer $n \geq 1$, there is an n-variable polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that:

- $p(1, 1, \ldots, 1) = 1$,
- $-1/3 \leq p(x) \leq 1/3$ for all $x \in \{0, 1\}^n$ other than $x = (1, 1, \ldots, 1)$, and
- the degree of p is $O(\sqrt{n})$.

3 APSP

In this problem, we will design an algorithm for APSP on unweighted, undirected graphs. Classic algorithms like Floyd-Warshall solve this problem in $O(n^3)$ time, but we will solve it faster!

Let G be an undirected, unweighted graph on n nodes. Let G_2 denote the graph on n nodes where (i, j) is an edge in G_2 if and only if there is a path of length at most 2 from i to j in G. Let $d(i, j)$ denote the distance from i to j in G, and $d_2(i, j)$ denote the distance from i to j in G_2.

(a) Prove that for all nodes i and j, we have $d_2(i, j) = \lceil d(i, j)/2 \rceil$.

(b) For a node i, let $N(i)$ denote the set of nodes adjacent to i in G. Prove that if i and j are nodes such that $d(i, j)$ is even, then for every $w \in N(i)$, we have $d_2(w, j) \geq d_2(i, j)$.

(c) Prove that if i and j are nodes such that $d(i, j)$ is odd, then for every $w \in N(i)$, we have $d_2(w, j) \leq d_2(i, j)$.

(d) Prove that if i and j are nodes such that $d(i, j)$ is odd, then there is a node $w \in N(i)$ such that $d_2(w, j) < d_2(i, j)$.

(e) Suppose we have already computed $d_2(i, j)$ for all nodes i and j. Show how to compute $d(i, j)$ for all nodes i and j in time $O(n^{2.373})$. Hint: Multiply $D \times A$, where D is the matrix with $D[i, j] = d_2(i, j)$, and A is the adjacency matrix of G.

(f) Give an algorithm which runs in time $O(n^{2.373} \log n)$ which, given as input an unweighted undirected graph G, computes $d(i, j)$ for all pairs of nodes i and j.

4 Detecting Subgraphs

Note: you may want to wait until after the class on September 21 to work on this problem.

Let A be the graph which looks like $\rhd\!\!-$, which has four nodes, and whose edges consist of a triangle plus one additional edge. Let K be the clique on 4 nodes (the graph with 4 nodes and all 6 edges). Recall the difference between a subgraph and an induced subgraph; for instance, A is a subgraph of K, but not an induced subgraph.

(a) Give a randomized algorithm with success probability $\geq 2/3$ which takes as input an n-node undirected graph G, and determines in $O(n^{2.373})$ time whether G contains A as an induced subgraph.

(b) Give a deterministic algorithm which takes as input an n-node undirected graph G, and determines in $o(n^4)$ time whether G contains K as an induced subgraph.