Exercise

Problem 1. Let $d, k \geqslant 1$ be fixed constants. Give a linear time algorithm for the following problem.

Instance:	A graph of degree bounded by d.
Problem:	Decide whether G has an independent set of size k.

Problem 2. Let $k \geqslant 1$ be a fixed constant. Give an $O(mn)$-time algorithm for the following CLUSTER EDITING problem, where m is the number of edges and n is the number of vertices:

Instance:	A graph G.
Problem:	Decide whether one can add or delete at most k edges in G so that it becomes a disjoint union of cliques.

Problem 3. A hypergraph $H = (V, E)$ consists of a set V of vertices and a set E of hyperedges, where each $e \in E$ is a subset of V. Let $d, k \geqslant 1$ be fixed. Give a quadratic time algorithm for the following problem.

Instance:	A hypergraph $H = (V, E)$ with $\|e\| \leqslant d$ for all $e \in E$.
Problem:	Decide whether there exists a subset $S \subseteq V$ with $\|S\| = k$ such that $e \cap S \neq \emptyset$ for every $e \in E$.

Hint: Use the Sunflower Lemma of Erdös and Rado.

Problem 4. Let χ be a quadratic residue character in $\mathbb{F}_q$: $\chi(x) = x^{(q-1)/2}$. That is, $\chi(x) = 1$ if x is a non-zero square in $\mathbb{F}_q$, $\chi(x) = -1$ if x is anon-square, and $\chi(0) = 0$. Also, $\chi(0) = 0$. Also, $\chi(x \cdot y) = \chi(x) \cdot \chi(y)$.

Theorem 0.1 (Weil1948)**.** *Let $f(t)$ be a polynomial over $\mathbb{F}_q$ which is not the square of another polynomial, and has precisely s distinct zeros. Then*

$$|\sum_{x \in \mathbb{F}_q} \chi(f(x))| \leqslant (s-1)\sqrt{q}.$$

Let $G = (V, E)$ be a graph with $V = \mathbb{F}_q$. Two vertices $x, y \in V$ are adjacent if $\chi(x + y) = 1$. For any $a, b \in V$, let $\mathrm{codeg}(a, b)$ be the number of vertices joined to a and b. Prove that $|\mathrm{codeg}(a, b) - q/4| \leqslant O(\sqrt{q})$ for large q.

Problem 5. The r-partite Turán problem is as follows. An r-partite graph is a graph whose vertices can be divided into r disjoint and independent sets $V_1, V_2, \cdots, V_r$ such that every edge connects a vertex in V_i to one in V_j where $i \neq j$. Formally, an r-partite graph $G = (V_1, V_2, \cdots, V_r, E)$, where $V_1, V_2, \cdots, V_r$ are the vertex sets and E is the edge set, satisfies

– $\forall i \neq j$, $V_i \cap V_j = \emptyset$;

– if $(u, v) \in E$ and $u \in V_i$ for some $i \in [r]$, then $v \notin V_i$.

Assume that there are at least $\rho|V_i||V_j|$ edges between each pair (V_i, V_j). Then at what density ρ must G contain K_r, a clique of size r?

(1) calculate ρ with the general Lovasz local lemma;

(2) calculate ρ with the cluster expansion lemma;

(3) calculate ρ with Shearer's lemma.

Hint: the line graph of an undirected graph G is another graph $L(G)$ constructed in the following way: for each edge in G, make a vertex in $L(G)$; for every two edges in G that have a vertex in common, make an edge between their corresponding vertices in $L(G)$. Given $L(K_r)$ and $\vec{p} = (p, \cdots, p)$, one can verify that $\check{q}_S(\vec{p}) \triangleq \sum_{\text{indep.} I \subseteq S} (-1)^{|I|} p^{|I|} > 0$ for each subset S of the vertices of $L(K_r)$ if $p < \frac{1}{4(r-2)}$.

Problem 6. For $N = 2^n$, we are given $x \in \{0,1\}^N$ with the Hamming weight t. The goal is to find an i such that $x_i = 1$.

(1) Prove that any randomized algorithm needs $\Theta(\frac{N}{t})$ queries to solve the problem with probability of success at least $2/3$

(2) If $t = N/4$, the Grover's algorithm always finds a solution with certainty after just one query.

Problem 7. Let $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

(1) Prove that any 2×2 matrix can be written as a linear combination of I, X, Y, Z.

(2) Let ρ be a density operator, with an expansion $\rho = c_0 I + c_1 X + c_2 Y + c_3 Z$. Prove that $c_0 = \frac{1}{2}$ and $c_i \in \mathbb{R}$. Moreover, $\sum_{i=1}^{3} c_i^2 \leqslant \frac{1}{4}$. The equality holds if and only if ρ is a rank-one matrix.

Problem 8. Consider a network containing n computers, some of these computers are connected by cables. Therefore, we can use an undirected graph $G = (V, E)$ to model the network, where $|V| = n$ and there is an edge between two vertices iff there is a cable connecting the two corresponding computers. Assume G is connected. For each edge $e \in E$, we associate a weight $w_e \in \mathbb{R}^+$ with it.

Initially, computers in the network do not know the topology of the network. That is, they do not know how G looks like. However, for each computer, it does know the edges incident to it (i.e., the cables connected to it), and the weights of these edges.

Two computers can communicate with each other if there is a cable connecting them. Specifically, the system evolves over rounds. At the beginning of each round, for each computer, for each of the cable connected to it, it can send a message to the computer at the other end of the cable. By the end of this round, for each computer u, for each of the cable connected to it, if the computer v at the other end of the cable sent a message $m_{v \to u}$ over this link at the beginning of this round, computer u will receive the message $m_{v \to u}$.

Now, suppose the computers want to compute a minimum spanning tree of G. Try to devise an algorithm and analyze how many rounds your algorithm needs. Also try to briefly argue the correctness of your algorithm.

Notice, your algorithm would be a *distributed* algorithm. That is, all computers in the network will run an identical algorithm, and these computers will run this algorithm in parallel. Furthermore, G should *not* be the input of your algorithm, since no computer knows the entire network topology initially. However, recall that each computer does know the cables connected to it and the weights of them. Moreover, computers can communicate with each other in each round to gradually gain more information regarding G.