- **Use of unfair means will be severely penalized.**

There are 6 questions for a total of 20 points.

---

1. Given a graph, counting the number of independent sets in the graph is called the # INDEPENDENT SETS problem. Show the following:
   (1) (a) Show that # INDEPENDENT SETS is #P-complete.
   (2) (b) Show that approximating # INDEPENDENT SETS within any constant factor is NP-hard.

(2) 2. Let GI denote the graph isomorphism problem. Show that if there is an efficient algorithm that solves the GI problem on $(n-1)$ vertex graphs, then there is an efficient algorithm that solves the problem on n vertex graphs.
(*Recall that this is known as the downwards self reducibility property*)

(3) 3. Consider the following optimization problem:

> Let V be a $\mathsf{PCP}(\log n, 1)$ verifier for 3SAT. On input ψ, a 3SAT formula, find a proof that maximises the acceptance probability of V.

Argue that there is no PTAS for the above problem, unless $\mathsf{P} = \mathsf{NP}$.

4. Show the following:
   (1) (a) A language L is said to be in $\mathsf{PCP}_{1,s}[O(\log n), 2]$ is there is a PCP verifier with completeness 1 and soundness s and that uses exactly 2 queries and $O(\log n)$ random bits. Show that if $\mathsf{SAT} \in \mathsf{PCP}_{1,s}[O(\log n), 2]$ for some constant $0 < s < 1$, then $\mathsf{P} = \mathsf{NP}$.
   (3) (b) If $\mathsf{SAT} \in \mathsf{PCP}(r(n), 1)$ for $r(n) = o(\log n)$ then $\mathsf{P} = \mathsf{NP}$.

(3) 5. For $b_1, b_2, b_3 \in \{0, 1\}$, let $\mathsf{MAJ}(b_1, b_2, b_3)$ denote the majority bit in b_1, b_2, b_3. Consider the following optimization problem:

> MAX-MAJ: Given a bunch of majority constraints over n Boolean variables $x_1, ..., x_n$ each containing 3 variables. That is, constraints of the form $\mathsf{MAJ}(x_{i_1}, x_{i_2}, x_{i_3}) = b$, find an assignment that satisfies the maximum number of such majority constraints.

Show that there does not exist an efficient $(2/3 + \epsilon)$ approximation for MAX-MAJ unless $\mathsf{P} = \mathsf{NP}$.

(5) 6. Let $w = H(x)$ be the Hadamard code for a message $x \in \{0, 1\}^n$. You receive a corrupted version $\tilde{w}$ of the codeword such that the following holds:

$$\mathbf{Pr}_{r \in \{0,1\}^n}[w[x] = \tilde{w}[x]] \geq 1/2 + \epsilon, \text{for some } \epsilon > 0$$

Show that there is an algorithm A that outputs the message x with probability at least $\Omega(\epsilon^2)$. Discuss the running time of your algorithm.
(*This problem involves reading and understanding the Goldreich-Levin Theorem. Please write the answer in your own words.*)