

1. For this problem, you should use the Chernoff bounds $\Pr[B(n, p) \leq (1 - \delta)np] \leq e^{-\delta^2 np/2}$ and $\Pr[B(n, p) \geq (1 + \delta)np] \leq e^{-\delta^2 np/3}$.

- (a) Say we throw a fair, six-sided die n times, and X is the number of times that the result is a 1. We are interested in the probability that $X \geq n/4$. Compare the best upper bounds on this probability you can obtain using Markov's inequality, Chebyshev's inequality, and Chernoff bounds.
- (b) A manufacturer produces chips that are faulty independently with probability p . They can test any chip to see if it is faulty, but would like to determine the value p without testing every chip. In order to do so, they use the following procedure: choose n chips uniformly and independently at random, and let X be the number of faulty chips encountered. Output X/n . We want to determine how large n should be so that we can be reasonably confident of the result. In particular, for a given ϵ and δ , $0 < \epsilon, \delta < 1$, we want to find a value N such that if $n \geq N$, then

$$\Pr[|X/n - p| \leq \epsilon p] \geq 1 - \delta.$$

Use the Chernoff bounds above to derive the smallest expression you can for a value of N such that this requirement is satisfied. Your expression for N should be in terms of ϵ , δ , and p .

2. Let $S = \{s_1, \dots, s_k\}$ be a set of finite binary strings. We say that a string x is a *concatenation over S* if it is equal to $s_{i_1} s_{i_2} \dots s_{i_t}$, where for each i_j , $1 \leq i_j \leq k$. Note that the same string from S can appear more than once in the concatenation. Consider the Concatenation-Collision problem: given two sets of finite binary strings A and B , does there exist any x that is a concatenation over both A and B ? Show that the Concatenation-Collision problem is in **NP**.

(Hint: the witness you use should be a string x that is a concatenation over both sets. Keep in mind, however, that a witness must have size polynomial in the size of the input.)

3. An independent set in a graph is a subset S of the nodes such that no pair of nodes in S have an edge between them. Finding the largest independent set in a graph is equivalent to finding the largest clique in the compliment graph, and thus determining if a graph has an independent set larger than a given k is **NP-Complete**.

A *strong independent set* is a subset S of the nodes such that no pair of nodes in S have either an edge, or a path of length two between them. In the decision version of the strongly independent set problem, you are given a graph G and an integer k , and you want to know if G has a strongly independent set of size at least k .

Show that the strongly independent set problem is **NP-Complete**.

4. A *Monotone CNF* formula is a Boolean formula in Conjunctive Normal Form, where all literals are non-negated. Thus $(x_1 \vee x_2) \wedge (x_2 \vee x_3 \vee x_4)$ is a Monotone CNF formula, but $(\bar{x}_1 \vee x_2) \wedge (x_2 \vee \bar{x}_3 \vee x_4)$ is not. All Monotone CNF formulas are satisfiable, since we can simply set all the variables to True. However, satisfying assignments where some of the variables are set to False can also exist.

In the Min-true Monotone CNF problem, we are given a Monotone CNF formula Φ and an integer k , and we want to know if Φ has a satisfying assignment with at most k variables set to True. Show that the Min-true Monotone CNF problem is **NP-Complete**.