Instructions

- You should think about each problem by yourself for at least an hour before choosing to collaborate with others.
- You are allowed to collaborate with fellow students taking the class in solving the problems (in groups of at most 2 people for each problem). But you must write your solution on your own.
- You are not allowed to search for answers or hints on the web. You are encouraged to contact the instructor or the TAs for a possible hint.
- You cannot collaborate on Extra credit problems
- Solutions typeset in LATEX are preferred.
- Feel free to use the Discussion Board or email the instructor or the TA if you have any questions or would like any clarifications about the problems.
- Please upload your solutions to Canvas. The solution to each problem must be uploaded separately.

In solving these assignments and any future assignment, feel free to use these approximations:

\[ 1 - x \approx e^{-x}, \quad n! \approx (n/e)^n, \quad \left( \frac{n}{k} \right)^k \leq \left( \frac{en}{k} \right)^k \]

Also, recall Cauchy-Schwartz Inequality: For real numbers \( a_1, \ldots, a_n, b_1, \ldots, b_n \) we have

\[ \sum_{i=1}^{n} a_i b_i \leq \sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2} \]

1) Let \( Y \) be a non-negative integer valued random variable. Prove the following inequalities:

\[ \frac{\mathbb{E}[Y]^2}{\mathbb{E}[Y^2]} \leq \mathbb{P}[Y \neq 0] \leq \mathbb{E}[Y] \]

2) a) Show how to construct a biased coin, which is 1 with probability \( p \) and 0 otherwise, using \( O(1) \) random bits in expectation. [Hint: First show how to construct a biased coin using an arbitrary number of random bits. Then show that the expected number of bits examined is small.]

b) Given \( p_1, \ldots, p_n \) where \( \sum_i p_i = 1 \), show how to sample from \( \{1, \ldots, n\} \) where \( i \) must be chosen with probability \( p_i \), using \( O(\log n) \) random bits in expectation.

c) Show that the “in expectation” caveat is necessary: for example, one cannot sample uniformly over \( \{1, 2, 3\} \) using \( O(1) \) bits in the worst case.

3) Let \( S = \{1, \ldots, n\} \) and \( T = \{n+1, \ldots, 2n\} \). Choose a random set \( R \) where each number \( 1, \ldots, 2n \) is in \( R \), independently, with probability \( p \).

a) Show that for \( p = 1/n \) with a constant probability (independent of \( n \)), \( R \cap S = \emptyset \) and \( R \cap T \neq \emptyset \).
b) Now assume that we choose elements of \( R \) only with a pairwise independent hash function, while still every element is chosen with probability \( p \). Choose a specific value of \( p \) (as a function of \( n \)) such that still with a constant probability (independent of \( n \)), \( R \cap S = \emptyset, R \cap T \neq \emptyset \).

4) Consider an \( n \)-dimension hypercube as a network of parallel processors. The network has \( N = 2^n \) processors where each processor is represented by an \( n \) bit string \( x_0x_1 \ldots x_{n-1} \) and two processors are connected by a wire if their bit representations differ in exactly one bit. We consider the permutation routing problem on such a network. Each processor \( x \) initially contains one packet \( p_x \) destined for some processor \( d(x) \) in the network such that each processor is the destination of exactly one packet, i.e., \( d(.) \) is a permutation. All communication between processors proceeds in a sequence of synchronous steps. At each time step each wire can transmit a single packet in each direction. So, in each step, a processor can send at most one packet to each of its neighbors.

We want to design an algorithm to specify a route for each packet, i.e., a sequence of edges from the source to the destination. Note that a packet may have to wait for several steps at an intermediate node \( y \) because multiple packets may want to leave \( y \) through the same wire. The goal is to design an algorithm to route all packets in a small number of steps.

(a) Consider the following simple strategy called `bit-fixing`. To send a packet \( p_x \) from node \( x \) to the node \( d(x) \), scan the bits of \( d(x) \) from left to right, and compare them with the address of the current location of \( p_x \), send \( p_x \) out of the current node along the edge corresponding to the left-most bit in which the current position and \( d(x) \) differ. For example, in going from 1011 to 0000 in a 4-dimensional hypercube, the packet would go through the pass 1011 \( \rightarrow \) 0011 \( \rightarrow \) 0001 \( \rightarrow \) 0000. Construct a permutation \( d(.) \) and prove that for such a permutation the bit-fixing strategy takes (at least) \( \Omega(\sqrt{N}/n) \) steps.

**Hint:** One way to prove such a lower bound is to find a node that at least \( \sqrt{N} \) packets will pass through it.

Now, consider the following 2-phase simple strategy. Pick a uniformly random intermediate destination \( \sigma(x) \) for each packet \( p_x \). In the first phase use bit-fixing to send \( p_x \) to \( \sigma(x) \). In the second phase send \( p_x \) from \( \sigma(x) \) to \( d(x) \). We prove that this routing strategy takes only \( O(n^2) \) steps\(^1\).

(b) Show that for each node \( y \) the expected number of packets that pass through \( y \) in the first phase is \( O(n) \).

(c) Use the Bernstein’s inequality to show that for each node \( y \) the number of packets that pass through \( y \) in the first phase is \( O(n) \) with probability at least \( 1 - 1/N^2 \).

**Theorem 1.1** (Bernstein’s inequality). Let \( X_1, \ldots, X_n \) be independent Bernoulli random variables. Then

\[
\mathbb{P} \left[ \sum_{i=1}^{n} X_i - \mathbb{E} \sum_{i=1}^{n} X_i > \epsilon \right] \leq \exp \left( \frac{-\frac{1}{2} \epsilon^2}{\sum \operatorname{Var}(X_i) + \epsilon/3} \right).
\]

(d) Prove that the 2-phase strategy takes only \( O(n^2) \) steps w.h.p..

5) In this problem you are supposed to implement min-cut Algorithm-1 and output the probability that it returns a min-cut of the given graph (note that in class we proved a lower bound of \( 1/(\binom{n}{2}) \) but the probability can be significantly larger) within 0.01 error.

I will uploaded three input files to the course website. Each file contains the list of edge of a graph; note that the graphs may also have parallel edges. The label of each node is an integer. For example, given the following input you should output 0.50. This graph has 4 edges and nodes have labels 1, 3, 4, 6. It

\(^1\)We remark that it is also possible to prove that the 2-phase strategy takes only \( O(n) \) steps but here we prove a weaker bound.
has a unique minimum cut which is the degree cut of vertex 1 and the probability that Algorithm 1 finds this cut is 0.50.

For each input file you should output the size of the mincut together with probability that algorithm-1 returns a mincut. Please upload your code to Gradescope and its output output of your program for each input in the designated “text box”.

6) **Extra Credit:** Say we have a plane with \( n \) seats and we have a sequence of \( n \) passengers 1, 2, \ldots, \( n \) who are going to board the plane in this order and suppose passenger \( i \) is supposed to sit at seat \( i \). Say when 1 comes he chooses to sit at some arbitrary seat different from his own sit, 1. From now on, when passenger \( i \) boards, if her seat \( i \) is available she sits at \( i \), otherwise she chooses sits at a uniformly random seat that is still available. What is the probability that passenger \( n \) sits at her seat \( n \)?