CS 473 ✧ Fall 2024
Midterm 2 Problem 1 Solution

Suppose you have access to a source that generates independent random bits, each equal to 1 with probability p. Consider the following game: In each round, you generate two bits using your random source and compute the AND of those two bits. If the AND is 1, the game immediately ends; otherwise, you continue to the next round. Let X denote the number of ***rounds*** until the game ends.

(a) What is the exact value of $\mathrm{E}[X]$?

(b) What is the exact value of $\mathrm{E}[X^2]$? *[Hint: Use a recursive definition of X.]*

Prove that both your answers are correct.

Solution (via recursion):

(a) The game always lasts for at least one round. The first round ends the game if and only if both of the first two bits are 1, which happens with probability p^2. Thus, with probability $1-p^2$, the game starts over after the first round. We conclude that

$$\mathrm{E}[X] = 1 + (1-p^2)\cdot \mathrm{E}[X] \implies \mathrm{E}[X] = \boxed{\frac{1}{p^2}}$$

(b) We use the following recursive characterization: $X = 1$ with probability p^2, and $X = 1 + Y$ with probability $1-p^2$, where Y is another random variable with exactly the same distribution as X. Thus,

$$\begin{aligned}
\mathrm{E}[X^2] &= p^2 \;+\; (1-p^2)\cdot \mathrm{E}[(1+Y)^2] \\
&= p^2 \;+\; (1-p^2)\cdot \mathrm{E}[(1+X)^2] && [X \text{ and } Y \text{ have same dist.}] \\
&= p^2 \;+\; (1-p^2)\cdot \mathrm{E}[1+2X+X^2] && [\text{math}] \\
&= p^2 \;+\; (1-p^2)\cdot \big(1+2\,\mathrm{E}[X]+\mathrm{E}[X^2]\big) && [\text{linearity}] \\
&= p^2 \;+\; (1-p^2) \;+\; (1-p^2)\cdot 2\,\mathrm{E}[X] \;+\; (1-p^2)\,\mathrm{E}[X^2] && [\text{math}] \\
&= p^2 + (1-p^2) + \frac{2(1-p^2)}{p^2} \;+\; (1-p^2)\,\mathrm{E}[X^2] && [\text{part (a)}] \\
&= \frac{2-p^2}{p^2} \;+\; (1-p^2)\,\mathrm{E}[X^2] && [\text{math}]
\end{aligned}$$

It follows that

$$p^2\cdot \mathrm{E}[X^2] = \frac{2-p^2}{p^2} \implies \mathrm{E}[X^2] = \boxed{\frac{2-p^2}{p^4}}$$

■

Solution (via direct calculation):

(a) For each positive integer k, define an indicator variable X_k that equals 1 if and only if the game lasts for ***at least*** k rounds. Then the total number of rounds is $X = \sum_{k\geq 1} X_k$, which implies

$$\mathrm{E}[X] \;=\; \sum_k \Pr[X_k = 1] \;=\; \sum_{k\geq 1}(1-p^2)^{k-1} \;=\; \sum_{j\geq 0}(1-p^2)^j \;=\; \frac{1}{1-(1-p^2)} \;=\; \boxed{\frac{1}{p^2}}$$

(b) Again, let $X_k = 1$ if and only if the game lasts ***at least*** k rounds. For any k, we have $X_k - X_{k+1} = 1$ if and only if the game lasts exactly k rounds.

$$\begin{aligned}
X^2 &= \sum_{k\geq 1} k^2(X_k - X_{k+1}) \\
&= \sum_{k\geq 1} k^2 X_k - \sum_{k\geq 1} k^2 X_{k+1} \\
&= \sum_{k\geq 1} k^2 X_k - \sum_{k\geq 1} (k-1)^2 X_k \\
&= \sum_{k\geq 1} (2k-1) X_k \\
&= 2\sum_{k\geq 1} k X_k - X
\end{aligned}$$

Taking expectations, we find

$$\mathrm{E}[X^2] \;=\; 2\sum_{k\geq 1} k\cdot\Pr[X_k = 1] - \mathrm{E}[X] \;=\; 2\cdot\boxed{\sum_{k\geq 1} k\cdot(1-p^2)^{k-1}} - \frac{1}{p^2}$$

Let's call the boxed summation S and evaluate it separately.

$$\begin{aligned}
S &= \sum_{k\geq 1} k\cdot(1-p^2)^{k-1} \\
&= \sum_{k\geq 1}(k-1)\cdot(1-p^2)^{k-1} + \sum_{k\geq 1}(1-p^2)^{k-1} \\
&= \sum_{j\geq 0} j\cdot(1-p^2)^j + \sum_{j\geq 0}(1-p^2)^j \\
&= (1-p^2)\sum_{j\geq 0} j\cdot(1-p^2)^{j-1} + \sum_{j\geq 0}(1-p^2)^j \\
&= (1-p^2)\cdot S + \frac{1}{p^2}
\end{aligned}$$

It follows that $p^2 S = 1/p^2$, and thus $S = 1/p^4$. Plugging this value back into our earlier expression, we conclude that

$$\mathrm{E}[X^2] = \boxed{\frac{2}{p^4} - \frac{1}{p^2}}$$

■

Solution (via probability knowledge):

(a) For any natural number k, we have $\Pr[X = k] = p^2(1-p^2)^{k-1}$. Thus, X is a geometric random variable with success probability p^2. It follows immediately that

$$\mathrm{E}[X] = \boxed{\frac{1}{p^2}}$$

(b) Recall that the variance of X is $\mathrm{Var}(X) = \mathrm{E}[X^2] - (\mathrm{E}[X])^2$. Because X is a geometric random variable with success probability p^2, its variance is $\mathrm{Var}(X) = \frac{1-p^2}{p^4}$. Our solution to part (a) implies $(E[X])^2 = \frac{1}{p^4}$. We conclude that

$$E[X^2] = (\mathrm{E}[X])^2 + \mathrm{Var}(X) = \frac{1}{p^4} + \frac{1-p^2}{p^4} = \boxed{\frac{2-p^2}{p^4}}$$

■

Rubric: 10 points = 5 for part (a) + 5 for part (b). This is more detail than necessary for full credit. These are not the only correct proofs.

CS 473 ✧ Fall 2024
Midterm 2 Problem 2 Solution

Let $U = \{1, 2, \ldots, n\}$ be a universe of n elements. Let $S_1, \ldots, S_m$ be m subsets of U, each with size $|S_i| = n/10$. We call another set of elements $X \subseteq U$ a *covering set* if X contains at least one element of each subset S_i. Suppose we randomly generate X by independently including each element of U with probability $p = (c \log m)/n$, for some constant $c \geq 1000$.

(a) ***Prove*** that for any fixed index i, the set X intersects S_i with probability at least $1 - 1/m^3$.

(b) ***Prove*** that X contains at most $c^2 \log m$ elements with probability at least $1 - 1/m^2$.

(c) ***Prove*** that X is a covering set of size $O(\log m)$ with probability at least $1 - 1/m$.

Solution: Let's assume that log means $\ln = \log_e$.

(a) Because each element of S_i is independently included or excluded from X, we have

$$\Pr[X \cap S_i = \varnothing] \;=\; (1-p)^{n/10} \;=\; \left(1 - \frac{c \log m}{n}\right)^{n/10} \;\leq\; e^{-(c \log m)/10}$$

by The World's Most Useful Inequality. Setting $\boldsymbol{c = 1000}$ gives us

$$\Pr[X \cap S_i = \varnothing] \;\leq\; e^{-100 \log m} \;= \frac{1}{m^{100}} \;\leq\; \frac{1}{m^3}.$$

(b) We immediately have

$$\mathrm{E}[|X|] \;=\; pn \;=\; c \log m,$$

and therefore

$$\Pr\big[|X| > c^2 \log m\big] \;=\; \Pr\big[|X| > c \cdot \mathrm{E}[|X|]\big]$$

Because $|X|$ is a sum of independent indicator variables, one for each element of U, the Chernoff bound $\Pr[|X| > (1+\delta)\mu] < e^{-\delta\mu/2}$ with $\delta = c - 1$ implies

$$\Pr\big[|X| > c \cdot \mathrm{E}[|X|]\big] \;\leq\; e^{-((c-1)c \log m)/2}$$

Finally, setting $\boldsymbol{c = 1000}$ gives us

$$e^{-((c-1)c \log m)/2} \;\leq\; e^{-499500 \log m} \;=\; \frac{1}{m^{499500}} \;\leq\; \frac{1}{m^2}.$$

(c) X is ***not*** a covering set of size at most $c^2 \log n$ if and only if either (1) $X \cap S_i = \varnothing$ for some index i, or (2) $|X| > c^2 \log m$. Parts (a) and (b) and the union bound imply that X is ***not*** a covering set of size at most $c^2 \log n$ with probability at most

$$\frac{m}{m^3} + \frac{1}{m^2} = \frac{2}{m^2} \leq \frac{1}{m}.$$

(The last inequality breaks down when $m = 1$, but in that csae the probability is *trivially* at most $1/m = 1$.) ■

Rubric: 10 points = 4 for part (a) + 4 for part (b) + 2 for part (c). This is more detail than necessary for full credit.

CS 473 ✧ Fall 2024
Midterm 2 Problem 3 Solution

You are planning an election in a city with n voters and m polling stations. You need to assign each voter to a single polling station where they can cast their vote. Each voter must be assigned a polling station within 5 miles of their residence, but at most 1000 voters can vote at any single polling station.

You have access to a function $\textsc{CloseEnough}(i, j)$ that returns $\textsc{True}$ if voter i lives within 5 miles of polling station j, and returns $\textsc{False}$ otherwise, in constant time. Describe an algorithm that either finds a legal assignment of polling stations to voters, or correctly reports that no such assignment exists.

Solution (reduce to matching): First build a bipartite graph $G = (V \sqcup P, E)$ with one vertex for each voter and 1000 vertices $p_{j,1}, p_{j,2}, \ldots, p_{j,1000}$ for each polling station j, and an edge $v_i p_{j,k}$ if and only if $\textsc{CloseEnough}(i, j) = \textsc{True}$. Altogether, G has $n + 1000m = O(n + m)$ vertices and at most $1000mn = O(mn)$ edges.

Then find a maximum matching M in G in $O(VE)$ time, as described in class. If M has n edges, then for each edge $v_i p_{j,k}$ in M, assign voter i to polling station j. Otherwise, report that no legal assignment is possible.

The overall algorithm runs in $O(VE) = \boldsymbol{O((n + m)mn)}$ ***time***. ■

Solution (reduce to maximum flow): First build a flow network $G = (V, E)$ with four types of vertices

- a source vertex s,
- a vertex v_i for each voter i,
- a vertex p_j for each polling station j, and
- a target vertex t;

and three types of edges:

- an edge $s \to v_i$ for each voter i, with capacity 1,
- an edge $v_i \to p_j$ for each voter i and polling station j such that $\textsc{CloseEnough}(i, j) = \textsc{True}$, with capacity 1,
- an edge $p_j \to t$ for each polling station j, with capacity 1000.

Altogether G has $2 + n + m = O(n + m)$ vertices and at most $n + m + mn = O(mn)$ edges.

Then compute an integral maximum flow in G in $O(VE)$ time (using Orlin's algorithm). If the maximum flow value is less than n, report that no legal assignment is possible. Otherwise, for each edge $v_i \to p_j$ that carries positive flow, assign voter i to polling station j.

The overall algorithm runs in $O(VE) = \boldsymbol{O((n + m)mn)}$ ***time***. ■

Solution (faster flows (11/10)): We can actually make the previous algorithm (slightly) faster using off-the shelf Ford-Fulkerson instead of Orlin's algorithm. The value of the maximum flow in G is at most n, so Ford-Fulkerson runs in $O(E \cdot |f^*|) = O(mn^2)$ time. Everything else is faster, so the overall algorithm runs in $\boldsymbol{O(mn^2)}$ ***time***. ■

Rubric: 10 points: standard reduction rubric

CS 473 ✧ Fall 2024
Midterm 2 Problem 4 Solution

Suppose you are given a 4CNF formula with n variables $x_1, \ldots, x_n$ and $m \geq 100$ clauses.

(a) Suppose we independently assign each variable x_i to be TRUE or FALSE with equal probability. What is the exact expected number of ***unsatisfied*** clauses under this assignment?

(b) ***Prove*** that the probability that at least $m/16+1$ clauses are ***unsatisfied*** is at most $1-C/m$ for some constant C.

(c) Part (b) implies that under a random assignment, the number of ***satisfied*** clauses is at least $15m/16$ with probability at least C/m. Using this fact, describe an *efficient* randomized algorithm that *always* finds an assignment that ***satisfies*** at least $15m/16$ clauses, and analyze its expected running time.

Solution: Suppose we are given a 4CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where each C_j is a clause with four literals.

(a) For each index $1 \leq j \leq m$, let $X_j = 1$ if the random assignment does ***not*** satisfy the clause C_j and $X_j = 0$ otherwise. Each clause C_j if and only if all its literals are FALSE. Since each literal is TRUE or FALSE with equal probability, we have $\Pr[X_j = 1] = 1/16$.

Let $X = \sum_{j=1}^m X_j$ denote the number of ***unsatisfied*** clauses. The *expected* number of unsatisfied clauses is exactly

$$\mathrm{E}[X] = \sum_{j=1}^{m} \Pr[X_j = 1] = \boxed{\frac{m}{16}}$$

(b) Markov's inequality immediately implies

$$\begin{aligned}
\Pr[X \geq m/16+1] &\leq \frac{\mathrm{E}[X]}{m/16+1} \\
&= \frac{m/16}{m/16+1} && \text{[part (a)]} \\
&= 1 - \frac{1}{m/16+1} && \text{[math]} \\
&\leq 1 - \frac{1}{m/8} && [\tfrac{1}{1+t} \geq \tfrac{1}{2t}] \\
&= \boxed{1 - \frac{8}{m}} && \text{[math]}
\end{aligned}$$

(c) Our algorithm repeatedly generates and tests independent random assignments, until we find one that satisfies at least $15m/16$ clauses.

Each iteration of the algorithm takes $O(m+n)$ time to generate a random assignment and then check each clause by brute force. Part (b) implies that in each iteration, we generate a good assignment with probability at least $8/m$. Let Y denote the number of iterations required to find a good assignment; we have

$$\mathrm{E}[Y] \le 1 + (1 - 8/m)\,\mathrm{E}[Y] \implies \mathrm{E}[Y] = m/8.$$

Thus, our brute-force algorithm runs in $\boldsymbol{O(m(m+n))}$ ***expected time***. ■

Rubric: 10 points = 3 for part (a) + 3 for part (b) + 4 for part (c)