

CS 574: Randomized Algorithms, Fall 2026

Homework 2

Released: Tuesday, October 6, 2026

Due: Tuesday, October 20, 2026 at 11:59pm

Instructions & Course Policies:

- **Collaboration:** You may work in groups of up to three, but each student must write their solutions independently, in their own words. List all collaborators and external resources on the first page of your submission.
- **AI and other resources:** You are encouraged to use AI agents to help format your solutions, and may also consult them about the mathematics. Declare all textbooks, papers, websites, and AI tools consulted, and explain how you used them, in a footnote or bibliography.
- **Submission:** Submit a single typeset $\text{\LaTeX}$ PDF on Gradescope. Start each problem you submit on a new page.
- **Clarity and rigor:** Define your random variables and justify each use of independence, conditioning, and a tail inequality. You may use a preceding subpart's stated conclusion even if you have not proved it.
- **Grading:** There are four problems, each worth 100 points. **You have to solve only one problem of your choice**, for a total of 100 points. Think about all four: they are fair game for the midterm.

Conventions. Write $[r] = \{1, \dots, r\}$, and let $\ln$ denote the natural logarithm. All fresh random choices are independent of previous choices unless stated otherwise.

Lecture coverage. This assignment covers the six teaching lectures from September 11 through October 2 (excluding the September 23 midterm): limited independence and hashing (Problem 1), conditional expectations and discrepancy (Problem 2), repeated balls-into-bins processes (Problem 3), and random routing on a grid (Problem 4).

Problem 1: From Collisions to Perfect Hashing

[100 points]

We want to store a fixed set S of $n \geq 2$ distinct keys from a finite universe U , so that membership queries take worst-case $O(1)$ time. For every positive integer r , assume we can sample a hash function $h : U \rightarrow [r]$ satisfying

$$\mathbb{P}[h(x) = h(y)] \leq \frac{1}{r} \quad (x \neq y).$$

Such a family is called *universal*. Assume sampling, evaluating, and storing a hash function take $O(1)$ time or words, key comparisons take $O(1)$ time, and initializing an array of r cells takes $O(r)$ time. No mutual independence of the hash values is assumed.

- (A) [20 points] **Pairwise independence can hide a disaster.** Let $q \geq 3$ be prime. Write $\mathbb{F}_q = \{0, 1, \dots, q-1\}$ for the integers with addition and multiplication modulo q , and use these integers as both keys and bin labels. For $a, b \in \mathbb{F}_q$, define

$$h_{a,b}(x) = (ax + b) \bmod q.$$

Let $\mathcal{H} = \{h_{a,b} \mid a, b \in \mathbb{F}_q\}$. Choose A, B independently and uniformly from $\mathbb{F}_q$, and put $h = h_{A,B}$. This samples h uniformly from $\mathcal{H}$. You may take as a fact that, for any distinct keys x, y ,

$$\Pr_{h \in \mathcal{H}}[h(x) = u \text{ and } h(y) = v] = q^{-2} \quad \text{for all } u, v \in \mathbb{F}_q.$$

Thus any two distinct keys have independent, uniform hash values. Now store all q keys $x \in \mathbb{F}_q$ by placing each key x in bin $h(x)$, using the one sampled hash function h . Determine the distribution of the maximum bin load. Compare the probability that all keys land in one bin with that probability under fully independent hashing. Why is applying a fully independent balls-into-bins Chernoff analysis unjustified here?

- (B) **[20 points] A primary table with a quadratic budget.** Sample $h : U \rightarrow [n]$. Let $B_i = \{x \in S : h(x) = i\}$, $b_i = |B_i|$, and $Q = \sum_{i=1}^n b_i^2$. Express Q using the number of colliding unordered pairs, and prove $\mathbb{E}[Q] \leq 2n - 1$. Give an $O(n)$ -time test for $Q \leq 4n$. If we reject h and try again whenever this test fails, bound the expected number of trials and the probability that the first k trials all fail.
- (C) **[20 points] Secondary tables without collisions.** Fix an accepted primary table. For each nonempty bucket of size b , sample a fresh universal hash into b^2 cells, accepting it only if the bucket's keys have no collisions. Prove that a trial succeeds with probability at least $1/2$, even after conditioning on the entire primary construction. Analyze its expected construction time, including initialization and collision detection. Treat $b = 1$ explicitly.
- (D) **[25 points] Assemble the dictionary.** Describe the complete data structure, including how to answer a query for a key *not* in S . Prove worst-case $O(n)$ space, worst-case $O(1)$ query time, and expected $O(n)$ construction time. Explain why the space bound holds for every accepted construction, and why the algorithm is Las Vegas. Your construction-time argument must account for all secondary retries, not just the primary retries.
- (E) **[15 points] From expected time to a deadline.** Suppose the preceding analysis gives $\mathbb{E}[W] \leq Cn$ for construction work W , for a known absolute constant C . Given $0 < \delta < 1/2$, design a version that always stops after $O(n \log(1/\delta))$ work and returns either a correct dictionary or an explicit failure. Prove that failure has probability at most δ , while keeping worst-case $O(n)$ space. Do not assume a Chernoff bound for W .

Problem 2: Derandomization: Balancing and Satisfying

[100 points]

In parts (A)–(D), let $A = (a_{ji})$ be a real $m \times n$ matrix, with $m, n \geq 1$. Assume every row has positive squared norm $v_j = \sum_{i=1}^n a_{ji}^2$; identically zero rows can be discarded. We seek $x \in \{-1, +1\}^n$ making every row sum $D_j = \sum_i a_{ji}x_i$ small. Set

$$\Delta_j = \sqrt{2v_j \ln(4m)}, \quad \lambda_j = \frac{\Delta_j}{v_j}.$$

You may use $\cosh z = (e^z + e^{-z})/2 \leq e^{z^2/2}$. For the algorithmic analysis, use an idealized real-arithmetic model in which arithmetic, comparisons, square roots, exponentials, and logarithms each cost $O(1)$. No finite-precision implementation is requested.

- (A) [15 points] **A random coloring.** Choose all x_i independently and uniformly from $\{-1, +1\}$. Bound $\mathbb{E}[e^{\lambda D_j}]$ for any real λ , then show

$$\mathbb{P}[|D_j| \geq \Delta_j] \leq \frac{1}{2m}.$$

Conclude that all row bounds hold simultaneously with probability at least $1/2$. Identify exactly where independence is used.

- (B) [20 points] **A potential that detects failure.** For a complete sign vector, define

$$\Phi(x) = \sum_{j=1}^m e^{-\lambda_j \Delta_j} (e^{\lambda_j D_j} + e^{-\lambda_j D_j}).$$

Show $\mathbb{E}[\Phi] \leq 1/2$, and show that any violation $|D_j| \geq \Delta_j$ implies $\Phi \geq 1$. Why is minimizing the conditional expectation of this potential sufficient, even if we cannot efficiently compute the conditional probability of a violation?

- (C) **[25 points] Expose one sign at a time.** After fixing $x_1, \dots, x_k$, put $s_{j,k} = \sum_{i \leq k} a_{ji} x_i$. Derive the explicit formula

$$\Phi_k = \sum_{j=1}^m e^{-\lambda_j \Delta_j} (e^{\lambda_j s_{j,k}} + e^{-\lambda_j s_{j,k}}) \prod_{i=k+1}^n \cosh(\lambda_j a_{ji})$$

for the conditional expectation of Φ when the remaining signs are still independent and uniform. Prove that one of the two choices for x_{k+1} never increases Φ_k . Give a deterministic algorithm and prove its guarantee, including the strict versus non-strict inequalities at the end.

- (D) **[20 points] Make the computation efficient.** Show how to implement (C) in $O(mn)$ time and $O(m)$ auxiliary space, excluding the input matrix and output signs. In particular, avoid recomputing an entire suffix product for each candidate sign at each step. Explain why the divisions in your update rule are legitimate.
- (E) **[20 points] The same method for weighted MAX-3SAT.** Now consider a Boolean formula on n variables with m clauses. Each clause is a disjunction of exactly three literals on distinct variables and has a nonnegative weight w_j . Let $W = \sum_j w_j$. Give a deterministic algorithm finding an assignment that satisfies clauses of total weight at least $7W/8$. Derive the conditional expected satisfied weight after a partial assignment, distinguishing clauses already satisfied, clauses already falsified, and unresolved clauses. First obtain an $O(nm)$ -time algorithm, then show how variable-to-clause incidence lists reduce the time to $O(n + m)$. Count arithmetic on weights as constant time. Explain whether your conditional objective should increase or decrease, and compare it with the potential used for discrepancy.

Problem 3: Three Games with Balls and Bins

[100 points]

Fix $n \geq 2$ labeled bins. Initially there are n balls. In every round, each ball currently in play independently chooses a uniform bin; all bins are emptied before the next round. Choices are fresh in every round. A round in which the stopping condition is met still counts. Find upper and lower bounds on the *expected* number of rounds that agree up to constant factors. In part (C), obtain the sharper estimate requested there. You may use the bounded-differences inequality: if f depends on m independent choices and changing one choice changes f by at most one, then

$$\mathbb{P}[f \leq \mathbb{E}[f] - u] \leq \exp(-2u^2/m) \quad (u > 0).$$

- (A) **[30 points] Sticky balls.** After each throw, all balls in the same bin stick together and become one larger ball. Remove these new balls from the bins and throw them again. Stop when just one ball remains. If K_t is the number of balls after t rounds, compute $\mathbb{E}[K_{t+1} \mid K_t = k]$. Prove that the expected stopping time is $\Theta(n)$.
- (B) **[40 points] Tetris balls.** After each throw, sort the bins by decreasing load. Remove one ball from every nonempty bin; collect and rethrow all remaining balls. Stop when no balls remain. Sorting changes only the order in which the bins are displayed. Let M_t be the number of balls after t rounds. Prove that the expected stopping time is $\Theta(\log \log n)$ as $n \rightarrow \infty$. In your lower bound, justify why unusually collision-free rounds do not dominate the argument.
- (C) **[30 points] Coin-flip bins.** After each throw, independently flip a fair coin for every nonempty bin. On heads, remove *all* balls in that bin; on tails, keep all of them for the next throw. Stop when none remain. Prove that the expected stopping time is $\log_2 n + O(1)$. Explain why balls that landed together cannot be treated as independent survivors.

Problem 4: Random Interleaving on a Grid

[100 points]

Let $n \geq 2$. The vertices of an $n \times n$ grid are (i, j) , where $i, j \in [n]$ are the row and column. There are n packets, indexed by $i = 1, \dots, n$; packet i starts at $(i, 1)$. Choose a uniformly random permutation π of $[n]$. Packet i is destined for $(\pi(i), n)$.

Packet i needs $n - 1$ rightward steps and $d_i = |\pi(i) - i|$ vertical steps, all toward its destination row. Independently for each packet, choose a uniformly random ordering of these $n - 1 + d_i$ required steps, then follow that path. Every undirected edge has two separate directed channels, each carrying at most one packet per round. Each directed channel has a FIFO queue; a nonempty queue transmits its first packet each round. Packets arriving at a vertex during a round may enter an outgoing queue at the start of the next round. Buffers are unlimited. Let A_i be the round in which packet i arrives, and put $T = \max_i A_i$.

For a directed edge e , let L_e be the number of chosen paths that use e . All expectations below include both the random permutation and the independent random path orderings. Do not assume that different packets' edge-use events are independent.

- (A) [15 points] **Length of a random route.** How many possible step orderings can packet i choose, conditional on π ? Show that every chosen path has length $n - 1 + d_i$, and compute the expected sum of all n path lengths. Explain why the random ordering changes the path but not its length.
- (B) [25 points] **Expected load of one edge.** Prove $\mathbb{E}[L_e] \leq 2$ for every directed edge e . For a rightward edge, group possible source–destination pairs by their vertical displacement d and direction, and note that a path crosses each column boundary exactly once. For a vertical edge, group pairs by d and use symmetry to find the expected number of vertical steps in any one of the n gaps between or outside the $n - 1$ rightward steps. The bounds 2 for rightward edges and $1/2$ for either vertical direction are sufficient.
- (C) [20 points] **Two packets at one edge.** For a fixed directed edge e , let P_{ij} be a uniformly random interleaving from $(i, 1)$ to (j, n) , and put $p_{ij} = \mathbb{P}[e \in P_{ij}]$. Use the uniform-permutation rule and

independence of the path orderings, conditional on π , to prove

$$\mathbb{E} \left[\binom{L_e}{2} \right] = O(1).$$

Your proof should use the bound from (B) and should not treat the destination rows as independent random choices.

- (D) **[20 points] Count waiting at an edge.** Let W_e be the sum, over all packets, of the rounds they spend waiting to traverse directed edge e . Prove the deterministic bound $W_e \leq \binom{L_e}{2}$. Deduce that the expected sum of all packets' waiting times is $O(n^2)$.
- (E) **[20 points] Average delay and last arrival.** Show that the expected average arrival time satisfies

$$\mathbb{E} \left[\frac{1}{n} \sum_{i=1}^n A_i \right] = \Theta(n).$$

Then give a deterministic $O(n^2)$ upper bound on T by counting all packet transmissions. Explain why your bound on expected *average* arrival time alone does not imply $\mathbb{E}[T] = O(n)$. A tight bound on $\mathbb{E}[T]$ is not requested.

References

- [MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [MU17] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. 2nd ed. Cambridge University Press, 2017.