CS 574: Randomized Algorithms, Fall 2026
Homework 1

Due: Wednesday, September 23, 2026 at 11:59pm

Instructions & Course Policies:

- **Collaboration Policy:** You are encouraged to work in groups of up to three students. However, each student must write up their solutions completely independently in their own words. You must explicitly list all group members and external resources consulted on the first page of your submission.
- **PLEASE** use AI agents to help you format your homework solution. You can consult with an AI agents if you want to about the solution itself (but if you do, what is the topic of the homework, anyway?).
- **Submission:** Solutions must be typeset in LaTeX and submitted as a single PDF on Gradescope. Start each problem on a new page.
- **Clarity and Rigor:** Solutions must be self-contained, mathematically rigorous, and as simple as possible. Clearly define all sample spaces, random variables, events, and invariants used.
- **Outside Tools / AI Policy:** If you consulted any textbooks, research papers, websites, search engines, or AI coding/reasoning assistants, you must explicitly declare them in a dedicated footnote or bibliography entry and state how they were used.
- **Grading:** This problem set contains 4 problems, each worth 100 points, for a total of 100 points, since you can pick which one of these problems you want to solve. **You have to solve only one problem.** I would suggest thinking about all of them, as they are fair game for the midterm.

Problem 1: The Lazy Susan, the Blindfolded Diner, and the Alias Cafeteria

[100 points]

At the eccentric *Restaurant of Unpredictability*, diners are treated to culinary games that pit algorithmic strategy against devious adversaries, followed by an automated dessert buffet powered by discrete sampling.

Part A: The Lazy Susan Game (Adversary vs. Randomness). There are $n \geq 3$ covered dishes placed in a circle on a rotating Lazy Susan table, indexed clockwise as $0, 1, \dots, n-1$. Each dish hides either a spicy pepper (denoted by H) or a sweet dumpling (denoted by T). The table initially contains a mix of both items (at least one H and at least one T).

In each round:

(i) The diner, who is blindfolded and cannot see the dishes or the table, selects an arbitrary non-empty subset of indices $I \subseteq \{0, 1, \dots, n-1\}$ and flips the contents of those dishes ($\mathsf{H} \leftrightarrow \mathsf{T}$).

(ii) If all n dishes currently show the same item (all H or all T), a celebratory gong sounds, and the diner wins the meal!

(iii) Otherwise, a mischievous waiter silently rotates the Lazy Susan by an arbitrary shift: index j moves to $(j + s) \bmod n$ for some integer shift $s \in \{0, \dots, n-1\}$ chosen by the waiter. The diner is not told the shift s.

(A) **Deterministic Doom ($n = 3$):** Prove that for $n = 3$, any deterministic strategy for the diner can be completely defeated by the waiter (the diner will never win). Specifically, show that the waiter can always choose a rotation after each round so that the diner's next flip cannot make all dishes identical. (Assume the waiter knows the strategy in advance, and places the original dishes before the first move.)

(B) **The Power of Unpredictability ($n = 3$):** Suppose the diner switches to a randomized strategy: in each round, the diner picks a single index $J \in \{0, 1, 2\}$ uniformly at random and flips the dish at J. Prove that this randomized game terminates in finite expected time, and compute the *exact* expected number of rounds until the diner wins.

(C) **The Square Mystery ($n = 4$):** For $n = 4$, show that there exists a *deterministic* sequence of flips that guarantees the diner wins in at most 7 rounds, regardless of the initial configuration and regardless of the waiter's rotations! Describe the exact sequence of moves and prove that it always terminates within 7 rounds.

Part B: The Alias Cafeteria (Discrete Distribution Sampling). Having finished the meal, the diner enters the dessert hall. There are n distinct desserts. Dessert i is chosen with probability $p_i > 0$, where $\sum_{i=1}^{n} p_i = 1$. The cafeteria generates random orders using *Walker's alias method* [Wal77], which partitions the probability mass into n equal buckets of capacity $1/n$, each containing at most two dessert labels.

(D) **A Concrete Tasting Menu:** Consider $n = 4$ desserts with probabilities:

$$p_1 = \frac{1}{2}, \quad p_2 = \frac{1}{4}, \quad p_3 = \frac{1}{8}, \quad p_4 = \frac{1}{8}.$$

Trace the execution of the greedy alias algorithm from class using the three queues S (small), M (exact), and L (large). Specifically:

  (i) State the initial contents of S, M, L.
  (ii) For each of the 4 steps, specify which items are popped from the queues, which bucket is produced (specifying primary label, primary weight, and alias label), and which residual item is pushed back.
  (iii) Write out the final table of 4 buckets.

(E) **Structural Properties of the Alias Table:** Let $(p_1, \dots, p_n)$ be an arbitrary discrete probability distribution with $p_i > 0$ for all i.

  (i) Prove that in the alias table constructed by the greedy algorithm, at least one bucket contains only a single item (i.e., its primary item has fill exactly $1/n$, and has no alias).
  (ii) Define the ***alias graph*** $G = (V, E)$ on vertex set $V = \{1, \dots, n\}$, where a directed edge (s, ℓ) exists if and only if item ℓ serves as the alias in the bucket where item s was the small item. Prove that G is a directed forest (it contains no directed cycles).

(F) **Sampling with Only Fair Coins:** Suppose your computer does not have a real-valued random number generator *rand*(n), but only an ideal fair coin that generates independent unbiased random bits in $\{0, 1\}$. Suppose all probabilities p_i are dyadic rationals of the form $p_i = a_i/2^k$ for positive integers $a_i \geq 1$ and an integer $k \geq 1$.

  (i) Describe how to sample an index from the alias table using only fair coin flips.
  (ii) Prove that the expected number of fair coin flips needed to produce one exact sample is at most $k + 2$.

Problem 2: The Council of Feuding Academics: Expectation, Cuts, and Independence

[100 points]

The Faculty Senate of the Department of Arcane Algorithms is plagued by intense rivalries. There are n professors and m pairwise feuds. We model this by an undirected conflict graph $G = (V, E)$, where $|V| = n$ and $|E| = m$.

Part A: The Great Office Relocation (Max-Cut via Linearity). The Dean decides to split the department across two buildings, North Hall (N) and South Hall (S). A feud between professors u and v is ***pacified*** if u and v are placed in different buildings (i.e., the edge $\{u, v\}$ crosses the cut (N, S)).

(A) **Random Partition:** Suppose each professor is assigned to North Hall or South Hall independently by a fair coin flip (probability 1/2 each). Let X denote the number of pacified feuds.

   (i) Compute the expected number of pacified feuds $\mathbb{E}[X]$.

   (ii) Conclude that every graph $G = (V, E)$ contains a bipartite subgraph with at least $m/2$ edges.

(B) **Finding a Good Cut:** Let $Y = m - X$ be the number of unpacified feuds (edges whose endpoints reside in the same hall).

   (i) Compute $\mathbb{E}[Y]$ and apply Markov's inequality to show that $\mathbb{P}[X \geq m/2] \geq \frac{1}{m+1}$.

   (ii) Present a deterministic greedy algorithm that runs in $O(n+m)$ time and always produces a partition (N, S) satisfying $X \geq m/2$. (Hint: Place professors one by one, maximizing the number of pacified feuds at each step; this is equivalent to derandomizing the coin-flipping algorithm via conditional expectation).

Part B: Committee Pairing and Weighted Max-SAT. The Senate must form specialized teaching committees under logical constraints. Each constraint is represented as a boolean clause C_j over literals from variables $x_1, \ldots, x_n$. Each clause C_j has an importance weight $w_j > 0$. Let $W = \sum_{j=1}^{m} w_j$ be the total weight of all constraints.

(C) **Approximating Weighted Max-SAT:**

(i) Suppose each clause C_j contains k_j distinct literals. We assign each variable $x_i \in \{\mathsf{TRUE}, \mathsf{FALSE}\}$ independently with probability $1/2$. Express the expected total weight of satisfied clauses in terms of w_j and k_j.

(ii) Suppose every clause has length at least 2 (i.e., $k_j \geq 2$ for all j). Prove that there exists an assignment whose satisfied weight is at least $\frac{3}{4}W$.

(iii) Give an explicit example of an unweighted 2-SAT formula (where all clauses have size 2) such that no assignment can satisfy strictly more than $3/4$ of the clauses.

Part C: The Peacetime Committee (The Caro-Wei Theorem). The Dean must appoint a grievance committee $I \subseteq V$ such that no two committee members are in conflict (i.e., I is an ***independent set*** in G). For each professor $v \in V$, let $d_v = \deg(v)$ denote the number of feuds involving v.

Consider the following randomized algorithm:

(I) Pick a permutation π of the vertices V uniformly at random from all $n!$ permutations.

(II) Include vertex v in I if and only if v appears before all of its neighbors in π:

$$\pi(v) < \pi(u) \quad \text{for all } u \in N(v).$$

(D) **Independence & Marginal Probabilities:**

(i) Prove that the subset I returned by this algorithm is deterministically an independent set in G.

(ii) For a fixed vertex $v \in V$, prove that $\mathbb{P}[v \in I] = \frac{1}{d_v+1}$.

(E) **The Caro-Wei Bound [Car79, Wei81]:**

(i) Using linearity of expectation, prove that $\mathbb{E}[|I|] = \sum_{v \in V} \frac{1}{d_v+1}$.

(ii) Using the convexity of $f(x) = \frac{1}{x+1}$ (or Cauchy-Schwarz), prove that:

$$\mathbb{E}[|I|] \;\geq\; \frac{n}{\overline{d}+1} \;=\; \frac{n^2}{2m+n},$$

where $\overline{d} = 2m/n$ is the average degree of G. Conclude that every graph G contains an independent set of size at least $\frac{n}{\overline{d}+1}$.

Problem 3: QuickSort Under the Microscope: Near-Misses, Extremes, and Medians

[100 points]

At *Pivot & Peril Inc.*, software engineers rely on randomized sorting and selection routines. To optimize cache coherence and comparator branch prediction, they need a microscopic analysis of which pairs of elements actually get compared during QuickSort and QuickSelect.

Let $S = \{s_1, s_2, \dots, s_n\}$ be an array of n distinct real numbers, indexed in strictly increasing order: $s_1 < s_2 < \cdots < s_n$. Standard randomized QuickSort picks a pivot uniformly at random from the current subproblem, partitions the elements around the pivot, and recurses on the two subarrays.

Let $X_{ij} \in \{0, 1\}$ be the indicator variable that element s_i and element s_j (for $i < j$) are compared at some point during the execution of QuickSort.

Part A: Microscopic Analysis of Comparisons.

(A) **Neighborly Comparisons:** Two elements s_i, s_j are ***immediate neighbors*** if their sorted ranks differ by 1 (i.e., $j = i + 1$).

  (i) What is the probability that s_i and s_{i+1} are compared?

  (ii) Let $N_{\text{adj}} = \sum_{i=1}^{n-1} X_{i,i+1}$ be the total number of comparisons between immediate neighbors. Compute the exact value of $\mathbb{E}\left[N_{\text{adj}}\right]$.

(B) **Distant Strangers:** Two elements s_i, s_j are ***distant strangers*** if their rank difference is at least $n/2$ (i.e., $j - i \geq n/2$).

  (i) Prove that for any pair of distant strangers, $\mathbb{P}\left[X_{ij} = 1\right] \leq \frac{4}{n}$.

  (ii) Let

$$N_{\text{dist}} = \sum_{1 \leq i < j \leq n,\, j - i \geq n/2} X_{ij}$$

  be the total number of comparisons between distant strangers. Prove that $\mathbb{E}[N_{\text{dist}}] \leq n$. (Contrast this with the total expected number of comparisons $\mathbb{E}[C] \approx 2n \ln n$; why do distant comparisons contribute so little?)

Part B: Median-of-Three QuickSort. To avoid poor pivot choices, the engineers implement ***Median-of-Three QuickSort***: whenever a recursive subproblem contains $m \geq 3$ elements, the algorithm samples 3 elements a, b, c uniformly at random (without replacement) from the subproblem and selects their median as the pivot.

(C) **Pivot Quality & Concentration:** Let the sorted elements of the subproblem be $u_1 < u_2 < \cdots < u_m$. A pivot is ***central*** if its rank falls in the middle half of the subproblem, namely between $\lceil m/4 \rceil$ and $\lfloor 3m/4 \rfloor$.

(i) In standard QuickSort (single random pivot), what is the probability that the pivot is central in the limit as $m \to \infty$?

(ii) In Median-of-Three QuickSort, prove that the probability that the pivot is central in the limit as $m \to \infty$ is exactly $\frac{11}{16} = 0.6875$.

(iii) Briefly explain why increasing the probability of a central pivot reduces the expected recursion depth and the constant factor in the expected number of comparisons.

Part C: QuickSelect for the Extremes. Suppose we run randomized QuickSelect to find the *minimum* element of an array of n elements (i.e., the element of rank $k = 1$). At each step, a pivot is chosen uniformly at random; if the pivot is not s_1, the algorithm recurses on the subarray of elements smaller than the pivot.

(D) **Expected Comparisons for Minimum Selection:** Let $T(n)$ be the expected number of comparisons performed by QuickSelect to find the minimum of n elements, with $T(1) = 0$.

(i) Explain why $T(n)$ satisfies the recurrence:

$$T(n) \;=\; (n-1) + \frac{1}{n}\sum_{i=1}^{n} T(i-1), \quad \text{for } n \geq 2, \text{ with } T(0) = T(1) = 0.$$

(ii) Prove by induction (or by subtracting $nT(n) - (n-1)T(n-1)$) that the exact solution is:

$$T(n) \;=\; 2n - 2H_n,$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number.

(iii) Provide an alternative proof of this result using backward analysis and indicator variables:

- Show that the minimum s_1 is compared to s_j ($j > 1$) with probability $\frac{2}{j}$.
- Show that no two non-minimum elements s_i, s_j ($1 < i < j$) are ever compared during the execution.
- Sum the resulting probabilities to re-derive $T(n) = 2n - 2H_n$.

Problem 4: Chebyshev's Watchdog: Polling, Independence, and the Almost-Median

[100 points]

The Kingdom of Algorand is holding a constitutional referendum. The Royal Statistician has been tasked with conducting accurate pre-election polls and computing fast summary statistics for massive sensor networks without processing entire datasets.

Part A: Pairwise vs. Mutual Independence. Chebyshev's inequality states that for any random variable X with expectation μ and variance σ^2, $\mathbb{P}[|X-\mu| \geq t\sigma] \leq 1/t^2$. In class, we emphasized that variance is additive whenever the underlying random variables are merely *pairwise independent*.

(A) **A Classical Counterexample:** Let $X_1, X_2 \in \{-1, +1\}$ be two independent, unbiased random bits: $\mathbb{P}[X_1 = +1] = \mathbb{P}[X_1 = -1] = 1/2$, and similarly for X_2. Define $X_3 = X_1 X_2$.

- (i) Prove that the three random variables X_1, X_2, X_3 are ***pairwise independent***.
- (ii) Prove that X_1, X_2, X_3 are *not* mutually independent.
- (iii) Compute $\mathbb{V}[X_1 + X_2 + X_3]$. Does Chebyshev's inequality hold for $S = X_1 + X_2 + X_3$?
- (iv) Briefly explain why Chebyshev's inequality requires only pairwise independence, while exponential concentration bounds (such as Chernoff bounds) require full mutual independence.

Part B: Political Polling via Chebyshev. The Royal Statistician wishes to estimate the unknown fraction $p \in (0, 1)$ of voters who support the referendum. The statistician surveys a uniform random sample of m voters with replacement. Let $Y_i = 1$ if voter i supports the referendum, and 0 otherwise. The estimator is the empirical sample mean:

$$\hat{p} = \frac{1}{m} \sum_{i=1}^{m} Y_i.$$

(B) **Confidence and Sample Size:**

(i) Show that $\mathbb{E}[\hat{p}] = p$ and prove that $\mathbb{V}[\hat{p}] \leq \frac{1}{4m}$ for any value of $p \in (0, 1)$.

(ii) The Crown requires a margin of error of at most ± 2 percentage points ($\varepsilon = 0.02$) with confidence at least 95% (failure probability $\delta \leq 0.05$):

$$\mathbb{P}\big[|\hat{p} - p| \geq 0.02\big] \;\leq\; 0.05.$$

Using Chebyshev's inequality, calculate the minimum sample size m that guarantees this condition.

Part C: The Almost-Median via Sampling. Given an unsorted array $A = [a_1, \dots, a_n]$ of n distinct numbers (where n is astronomically large), an element $x \in A$ is said to be an ***ε-approximate median*** if its rank in the sorted order of A satisfies:

$$\left(\frac{1}{2} - \varepsilon\right) n \;\leq\; \mathrm{rank}(x) \;\leq\; \left(\frac{1}{2} + \varepsilon\right) n.$$

Consider the following sampling algorithm:

(I) Sample m elements $R = \{r_1, \dots, r_m\}$ uniformly at random from A with replacement, where m is odd.

(II) Sort the sample R and return M, the exact median of the sample (the element of rank $(m + 1)/2$ in R).

(C) **Chebyshev Analysis of the Sample Median:** Let $s_L \in A$ be the element of rank $\lfloor (1/2 - \varepsilon)n \rfloor$, and let $s_R \in A$ be the element of rank $\lceil (1/2 + \varepsilon)n \rceil$.

(i) Let Y_L be the number of sampled elements that are smaller than or equal to s_L. What is the distribution of Y_L? What are $\mathbb{E}[Y_L]$ and $\mathbb{V}[Y_L]$?

(ii) Prove that the sample median M is strictly smaller than s_L if and only if $Y_L \geq \frac{m+1}{2}$.

(iii) Show that $\frac{m+1}{2} - \mathbb{E}[Y_L] \geq \varepsilon m$.

(iv) Using Chebyshev's inequality, prove that if we choose $m = \lceil \frac{2}{\varepsilon^2} \rceil$, then:

$$\mathbb{P}[M < s_L] \;\leq\; \frac{1}{8} \quad \text{and} \quad \mathbb{P}[M > s_R] \;\leq\; \frac{1}{8}.$$

Conclude that with probability at least $3/4$, the sample median M is an ε-approximate median of A.

(D) **Independence of Dataset Size:** Observe that the sample size $m = O(1/\varepsilon^2)$ is *completely independent* of the total size n of the dataset.

(i) Why is this property of crucial importance in modern streaming algorithms and massive database systems where $n \approx 10^{12}$?

(ii) What is the total running time of the approximate median algorithm in terms of m, and how does it compare to the linear time $O(n)$ required for exact median selection?

Bibliography

[Car79] Y. Caro. New results on the independence number. *Technical Report, Tel-Aviv University*, 1979.

[Wal77] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3): 253–256, 1977.

[Wei81] V. K. Wei. A lower bound on the stability number of a graph. *Bell Laboratories Technical Memorandum*, 81-11217-9, 1981.