Homework 3

CSCI 4030 / CSCI 6220: Randomized Algorithms — Fall 2026

Rensselaer Polytechnic Institute — CSCI 4030: 100 points; CSCI 6220: 120 points

Randomized algorithms are useful because randomness can make systems faster, simpler, and more scalable—but it also creates fluctuations that we need to understand. In Lectures 5 and 6, balls-and-bins gave us a clean setting for several ideas that show up much more broadly: turning a tail bound for one random object into a guarantee about the worst of many objects, proving that a guarantee holds simultaneously over many times, and controlling an adaptive random process by comparing it to a simpler one.

This homework puts those ideas to work in settings that arise naturally in computer science. You will study stragglers in distributed computation, where one unusually slow task can determine the completion time of an entire job; online monitoring, where empirical frequencies must remain reliable as data arrive continuously; and sparse random networks, where a BFS exploration reveals whether local searches, failures, or contagion can spread very far. The goal is to see concentration, tail bounds, geometric checkpoints, truncation, and conditional reasoning as reusable tools for analyzing randomized systems—not as techniques tied to balls-and-bins.

Graduate students will also analyze Gaussian random projections for high-dimensional data. This problem shows how the same local-to-global concentration ideas lead to a practical randomized algorithm for dimensionality reduction and nearest-neighbor search: many high-dimensional distances can be compressed at once while still remaining accurate enough to support useful computations.

Justify every claim. A numerical answer without an argument earns no credit. Acknowledge any collaborators and any external sources you consult, including AI tools, and describe clearly what role they played in your submission.

Problems 1–3 are required of everyone and total 100 points. CSCI 6220 is graded out of 120 points and must also complete the 20-point Graduate Extension. The Graduate Extension is worth up to 20 points of extra credit for CSCI 4030.

Throughout the homework, log denotes the natural logarithm.

[30 points] Problem 1 (Stragglers and maxima of random quantities).

A distributed job is often split into many tasks that run in parallel. The job cannot finish until every required task is complete, so its completion time is the maximum of the task completion times. This is the *straggler problem*: even if one task is unlikely to be very slow, a system with many tasks has many chances to produce an unusually slow one.

Let $T_1, \dots, T_n \geq 0$ be the task completion times and

$$M_n = \max_{1 \leq i \leq n} T_i.$$

Unless stated otherwise, do not assume the T_i are independent. The same mathematics applies whenever a system is controlled by the largest of many random quantities, such as errors, delays, or noise coordinates.

(a) (6 points) For $t \geq 0$, let

$$Z_t = \sum_{i=1}^{n} \mathbf{1}\{T_i \geq t\}.$$

Prove

$$\mathbb{P}[M_n \geq t] \leq \mathbb{E}[Z_t] = \sum_{i=1}^{n} \mathbb{P}[T_i \geq t].$$

Your proof must not use independence. The useful move is to replace a maximum event by a count of threshold exceedances: control the maximum by controlling how many individual quantities cross the threshold.

(b) (9 points) Fix $\delta \in (0, 1)$ and assume $\tau, \sigma, q > 0$. In each of the first three cases, find an explicit $t = f(n, \delta)$ such that $\mathbb{P}[M_n \geq t] \leq \delta$, and also give the threshold when $\delta = 1/n$.

(i) $\mathbb{P}[T_i \geq t] \leq e^{-t/\tau}$ for every i.

(ii) $\mathbb{P}[T_i \geq t] \leq e^{-t^2/(2\sigma^2)}$ for every i.

(iii) $\mathbb{P}[T_i \geq t] \leq (1 + t/\tau)^{-q}$ for every i.

(iv) Set $\tau = \sigma = 1$, $q = 2$, $n = 10^4$, and $\delta = 1/n$. Compute the three thresholds numerically.

The numerical comparison in (iv) shows how strongly the tail shape controls the worst of many random quantities: light tails produce only logarithmic or square-root-logarithmic growth, while polynomial tails can produce much larger stragglers.

(c) (7 points) Now forget runtimes and let $X_1, \dots, X_n$ be arbitrary random variables satisfying the sub-Gaussian MGF bound

$$\mathbb{E}[e^{\lambda X_i}] \leq e^{\sigma^2 \lambda^2 / 2} \qquad \text{for every } \lambda > 0.$$

Let $X_{\max} = \max_i X_i$. The variables need not be independent.

Use

$$e^{\lambda X_{\max}} \leq \sum_{i=1}^{n} e^{\lambda X_i}$$

and Jensen's inequality to show that, for every $\lambda > 0$,

$$\mathbb{E}[X_{\max}] \le \frac{\log n}{\lambda} + \frac{\sigma^2 \lambda}{2}.$$

Optimize over λ to prove

$$\mathbb{E}[X_{\max}] \le \sigma\sqrt{2\log n}.$$

If $X_i \sim N(0, \sigma^2)$ independently, then σ^2 is the variance, and a standard sharper result is

$$\frac{\max_{i \le n} X_i}{\sigma\sqrt{2\log n}} \longrightarrow 1 \qquad \text{in probability.}$$

You are not asked to prove this sharper statement. It shows that the variance determines the basic fluctuation scale, while taking the maximum of n independent Gaussian coordinates introduces the $\sqrt{2\log n}$ factor.

(d) (8 points) Some random quantities have Gaussian-like moderate deviations but heavier far tails. A common one-sided sub-exponential condition is

$$\mathbb{E}[e^{\lambda X_i}] \le e^{\nu^2\lambda^2/2} \qquad \text{for } 0 \le \lambda \le \frac{1}{b}.$$

Here ν controls the variance-like scale of ordinary fluctuations, while b limits how far the Gaussian-like MGF behavior extends.

First use the MGF bound and Markov's inequality to prove the two-regime tail bound

$$\mathbb{P}[X_i \ge t] \le \begin{cases} \exp\left(-t^2/(2\nu^2)\right), & 0 \le t \le \nu^2/b, \\ \exp(-t/(2b)), & t > \nu^2/b. \end{cases}$$

Your proof should make clear why the optimizing value $\lambda = t/\nu^2$ is available in the first regime but hits the constraint $\lambda \le 1/b$ in the second.

Now let $X_{\max} = \max_i X_i$ and $L = \log(n/\delta)$. Use the tail bound you just proved to find a sufficient threshold $t = f(n, \delta, \nu, b)$ for

$$\mathbb{P}[X_{\max} \ge t] \le \delta.$$

Give the answer in two cases, determine the condition on L separating them, and express the result equivalently using the two terms $\nu\sqrt{L}$ and bL.

[30 points] Problem 2 (A request-frequency dashboard that stays accurate).

An online service receives one request at each time $t = 1, 2, \ldots$. Each request has one of d types: for example, API endpoints, query classes, or error codes. Let

$$X_t \in \{1, \ldots, d\}, \qquad \mathbb{P}[X_t = i] = p_i,$$

and assume the requests are independent and identically distributed.

After the first t requests, define

$$N_i(t) = \sum_{s=1}^{t} \mathbf{1}\{X_s = i\}, \qquad \widehat{p}_i(t) = \frac{N_i(t)}{t}.$$

Thus $N_i(t)$ counts type-i requests among the first t arrivals, and $\widehat{p}_i(t)$ is the empirical frequency shown by the dashboard.

Fix $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and $p_{\min} \in (0, 1]$. Our goal is to find a warm-up time m such that, with probability at least $1 - \delta$,

$$|\widehat{p}_i(t) - p_i| \le \varepsilon p_i$$

simultaneously for every type with $p_i \ge p_{\min}$ and *every* $t \ge m$. Ignore integer-rounding issues throughout this problem.

Algorithm 1: Continuous empirical-frequency monitor
Input: request stream $X_1, X_2, \ldots$ and number of types d.
State: counters $N_1, \ldots, N_d$, initialized to 0.
At time $t = 1, 2, \ldots$:

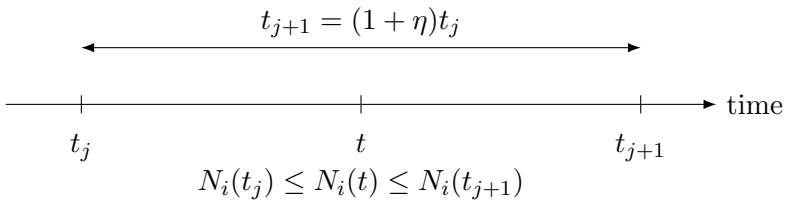
1. Observe X_t and increment N_{X_t}.
2. Display $\widehat{p}_i(t) = N_i(t)/t$ for each type i.

For this problem, you may use the following Chernoff bounds. If S is a sum of independent Bernoulli random variables with mean μ, then for $0 \le \eta \le 1$,

$$\mathbb{P}[S \ge (1+\eta)\mu] \le e^{-\eta^2\mu/3}, \qquad \mathbb{P}[S \le (1-\eta)\mu] \le e^{-\eta^2\mu/2}.$$

You may also use $1 - e^{-x} \ge x/2$ for $0 \le x \le 1$.

A guarantee at one fixed time does not certify a dashboard that may be read at any later time. We will compare two proofs of an infinite-horizon guarantee. The first sums the failure probabilities over every time. The second checks only geometrically spaced times and uses monotonicity of the counts to fill the gaps.



A multiplicative grid matches a relative-error guarantee: nearby checkpoints have nearly the same sample size, while monotonicity squeezes the count at every time between them.

(a) (6 points) Fix a type i and a time t. Prove that for $0 < \eta \leq 1$,

$$\mathbb{P}[|\widehat{p}_i(t) - p_i| > \eta p_i] \leq 2e^{-\eta^2 t p_i/3}.$$

(b) (7 points) First certify the dashboard by summing the bound in part (a) over every type and every time $t \geq m$. Show that there is a universal constant C such that

$$m \geq \frac{C}{\varepsilon^2 p_{\min}} \left(\log \frac{d}{\delta} + \log \frac{1}{\varepsilon^2 p_{\min}} \right)$$

is sufficient for the desired guarantee for all $t \geq m$. You do not need to determine C.

(c) (9 points) Now use geometric checkpoints. Set

$$\eta = \frac{\varepsilon}{4}, \qquad t_j = m(1+\eta)^j, \qquad j = 0, 1, 2, \ldots.$$

Fix a type i and suppose that at every checkpoint,

$$|\widehat{p}_i(t_j) - p_i| \leq \eta p_i.$$

Using

$$N_i(t_j) \leq N_i(t) \leq N_i(t_{j+1}) \qquad (t_j \leq t \leq t_{j+1}),$$

prove that

$$|\widehat{p}_i(t) - p_i| \leq \varepsilon p_i$$

for every $t \geq m$.

(d) (8 points) Use parts (a) and (c), together with a union bound over all d types and all checkpoints $j \geq 0$, to show that there is a universal constant C such that

$$m \geq \frac{C}{\varepsilon^2 p_{\min}} \left(\log \frac{d}{\delta} + \log \frac{1}{\varepsilon} \right)$$

is sufficient. You do not need to determine C.

For the probability sum, you may use

$$(1+\eta)^j \geq 1 + j\eta$$

to compare the checkpoint failure probabilities with a geometric series. Thus geometric checkpointing removes the extra dependence on $p_{\min}$ inside the logarithmic overhead while certifying the dashboard for all future times.

[40 points] Problem 3 (BFS in a sparse random network).

The Erdős–Rényi random graph $G(n,p)$ has vertex set $[n]$. Each of the $\binom{n}{2}$ possible edges is included independently with probability p.

In this problem,

$$p = \frac{c}{n}, \qquad 0 < c < 1.$$

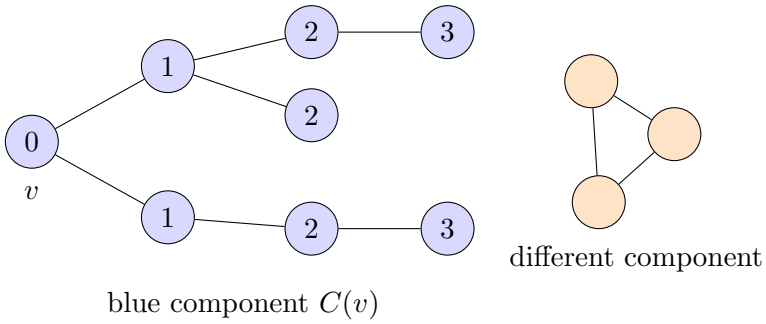
A vertex therefore has expected degree $(n-1)c/n \approx c < 1$. This is the *subcritical* sparse regime.

Our goal is to prove that, with high probability, every connected component has only $O(\log n)$ vertices. This matters whenever connectivity represents possible propagation: a failure, infection, or message can reach only its connected component. It also says that a local BFS or DFS query explores only a small part of the network. More broadly, this is the subcritical side of the Erdős–Rényi phase transition: below average degree 1, no giant connected cluster forms.

Algorithm 2: Explore the connected component of a start vertex
Input: graph $G = (V, E)$ and start vertex v.
State: discovered set S and FIFO queue Q.
Initialize: $S \leftarrow \{v\}$ and $Q \leftarrow [v]$.
While Q is nonempty:

1. Remove the first vertex u from Q.
2. Reveal every edge from u to a vertex in $V \setminus S$.
3. Add every newly discovered neighbor to S and append it to Q.

Output: $S = C(v)$, the connected component containing v.



BFS started at v discovers exactly $C(v)$. The numbers inside the blue vertices give their graph distance from v; vertices in the orange component are never reached. In the analysis below, however, one step means processing one queued vertex, and Y_t counts the new vertices discovered during that step.

Let Y_t be the number of new vertices discovered when BFS processes its tth vertex. Let $\mathcal{F}_{t-1}$ denote the entire exploration history revealed before BFS step t begins, including the discovered set and queue. Let D_{t-1} be the number of vertices already discovered at that moment. Ignore integer-rounding issues and set

$$k = \frac{12 \log n}{(1-c)^2}.$$

(a) (8 points) Condition on the entire BFS history before step t. If that step occurs, justify carefully using *deferred decisions* that every edge from the vertex processed at step t to an

undiscovered vertex is still an independent Bernoulli(c/n) random variable. Deduce that

$$Y_t \sim \mathrm{Bin}(U_t, c/n)$$

for some $\mathcal{F}_{t-1}$-measurable $U_t \le n$, and then show that for every $\theta > 0$,

$$\mathbb{E}[e^{\theta Y_t} \mid \mathcal{F}_{t-1}] \le \exp\Big(c(e^\theta - 1)\Big).$$

(b) (12 points) For $t = 1, \dots, k$, define

$$\widetilde{Y}_t = \begin{cases} Y_t, & \text{if step } t \text{ occurs and } D_{t-1} \le k \text{ when the step begins,} \\ 0, & \text{otherwise.} \end{cases}$$

Whether the increment is kept is therefore determined by $\mathcal{F}_{t-1}$, before the randomness of step t is revealed. Prove

$$\{|C(v)| \ge k+1\} \subseteq \left\{\sum_{t=1}^{k} \widetilde{Y}_t \ge k\right\}.$$

Then prove, using the tower property repeatedly, that

$$\mathbb{E}\left[\exp\left(\theta \sum_{t=1}^{k} \widetilde{Y}_t\right)\right] \le \exp\Big(kc(e^\theta - 1)\Big).$$

Do not assume the $\widetilde{Y}_t$ are independent. The truncation replaces a random-length adaptive process by a fixed number of increments whose conditional MGFs can still be controlled.

(c) (10 points) Set $\theta = (1-c)/2$.

(i) Prove

$$e^x - 1 \le x + x^2 \qquad \text{for } 0 \le x \le \frac{1}{2}.$$

(ii) Use this inequality to prove

$$\theta - c(e^\theta - 1) \ge \frac{(1-c)^2}{4}.$$

(iii) Deduce that

$$\mathbb{P}[|C(v)| \ge k+1] \le \exp\left(-\frac{k(1-c)^2}{4}\right) \le n^{-3}.$$

Thus the probability that BFS from one fixed machine reaches more than k machines is at most n^{-3}. Large connected regions are exponentially unlikely in the subcritical regime.

(d) (10 points) Let

$$L_n = \max_{v \in [n]} |C(v)|$$

be the size of the largest component.

The random variables $|C(v)|$ are highly dependent, but independence was not needed in Problem 1(a). Use the same threshold-counting argument and part (c) to prove

$$\mathbb{P}[L_n \ge k+1] \le n^{-2}.$$

Conclude that with probability at least $1 - n^{-2}$, every failure, infection, or message confined to graph edges can reach at most k vertices, and every connected-component search examines at most k vertices.

Optional bonus (2 extra-credit points). Before choosing θ in part (c), the same argument gives

$$\mathbb{P}[|C(v)| \geq k+1] \leq \exp\Big[-k\big(\theta - c(e^{\theta} - 1)\big)\Big] \qquad (\theta > 0).$$

Find the value of θ that maximizes the exponent and prove the sharper bound

$$\mathbb{P}[|C(v)| \geq k+1] \leq \exp\big[-k\big(c - 1 - \log c\big)\big] .$$

[20 points] Problem 4 (Graduate Extension: Gaussian random projections and approximate nearest-neighbor search).

This problem is required for CSCI 6220 and extra credit for CSCI 4030.

Suppose a database contains $N \geq 2$ vectors $x_1, \ldots, x_N \in \mathbb{R}^d$, where d may be very large. Exact distance computations cost $\Theta(d)$ per comparison. A Gaussian random projection replaces every vector by a vector in $\mathbb{R}^k$, with $k \ll d$, so that search can be carried out using the compressed representations.

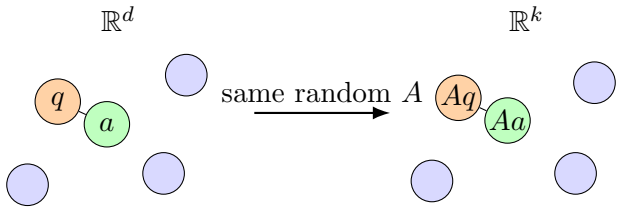
Algorithm 3: Gaussian random projection for nearest-neighbor search
Input: database vectors $x_1, \ldots, x_N \in \mathbb{R}^d$ and target dimension k.
Preprocessing:
1. Sample one matrix $A \in \mathbb{R}^{k \times d}$ with independent entries $A_{rs} \sim N(0, 1/k)$.
2. Store $y_i = Ax_i \in \mathbb{R}^k$ for every database vector.

Query: given $q \in \mathbb{R}^d$, compute Aq and return the index

$$\widehat{a} \in \arg\min_{1 \leq i \leq N} \|Aq - Ax_i\|_2.$$

Output: the database point $x_{\widehat{a}}$.

The question is how small k can be while preserving the distances needed by the search algorithm.



All points are compressed using the same random matrix. The goal is not to preserve every distance exactly, but to preserve the query-to-database distances well enough that the returned point is still a good nearest neighbor.

You may use the following Gaussian fact without proof. If $z \sim N(0, I_d)$ and $u \in \mathbb{R}^d$ is fixed, then

$$\langle z, u \rangle \sim N(0, \|u\|_2^2).$$

For a unit vector u, this is a standard normal random variable. This follows from the rotational symmetry of a multivariate Gaussian: no direction is distinguished.

You may also use, for $g \sim N(0, 1)$,

$$\mathbb{E}[e^{\lambda g^2}] = (1 - 2\lambda)^{-1/2}, \qquad \lambda < 1/2.$$

(a) (3 points) Fix a nonzero vector $x \in \mathbb{R}^d$. Write the rth row of A as $z_r^\top / \sqrt{k}$, where $z_r \sim N(0, I_d)$ and the z_r are independent. Using the Gaussian fact above, show that there are independent $g_1, \ldots, g_k \sim N(0, 1)$ such that

$$\|Ax\|_2^2 = \frac{\|x\|_2^2}{k} \sum_{r=1}^{k} g_r^2.$$

(b) (6 points) Use the supplied MGF and a Chernoff argument to prove that there is a universal constant $\gamma > 0$ such that, for every fixed $x \neq 0$ and $0 < \varepsilon < 1$,

$$\mathbb{P}\big[\big|\|Ax\|_2^2 - \|x\|_2^2\big| > \varepsilon\|x\|_2^2\big] \leq 2e^{-\gamma k\varepsilon^2}.$$

You do not need to determine γ. For the lower tail, apply the same MGF argument with a negative value of λ.

(c) (4 points) Fix $Q \geq 1$ query vectors $q_1, \ldots, q_Q$ *before* the matrix A is sampled. Apply part (b) to the NQ vectors $q_\ell - x_i$ and prove that there is a universal constant C such that

$$k \geq C\varepsilon^{-2} \log \frac{NQ}{\delta}$$

is sufficient to preserve all NQ query-to-database squared distances within a factor $1 \pm \varepsilon$ simultaneously, with probability at least $1 - \delta$. Independence among the failure events is not needed.

(d) (3 points) Fix one of the queries q. Let x_a be a true nearest neighbor of q, and let $x_{\widehat{a}}$ be the point returned by Algorithm 3. Assume all query-to-database squared distances are preserved within a factor $1 \pm \varepsilon$. Prove

$$\|q - x_{\widehat{a}}\|_2^2 \leq \frac{1+\varepsilon}{1-\varepsilon}\|q - x_a\|_2^2.$$

Thus the random projection gives an approximate nearest neighbor without any separation assumption between the best and second-best database points.

(e) (2 points) The guarantee in part (c) requires the queries to be fixed before A is sampled. Show why this matters. Assume $k < d$ and let an adversary see A before choosing a query. Prove that A has a nonzero vector v in its null space. For any database point x_1, consider the adaptive query

$$q = x_1 + v.$$

Show that

$$Aq = Ax_1 \qquad \text{but} \qquad \|q - x_1\|_2 > 0.$$

Conclude that no multiplicative distance-preservation guarantee of the form used in part (c) can hold simultaneously for every query chosen after seeing A.

(f) (2 points) Ignore the one-time cost of preprocessing the database. A brute-force exact query in $\mathbb{R}^d$ costs $\Theta(Nd)$. Show that Algorithm 3 costs

$$\Theta(kd + Nk)$$

per query: $\Theta(kd)$ to compute Aq and $\Theta(Nk)$ to compare with the projected database vectors. Derive the condition

$$k(d + N) = o(Nd)$$

under which projected search is asymptotically cheaper than exact brute-force search. In particular, verify that this holds when $k = o(\min\{N, d\})$.