

COS 330: Great Ideas in Theoretical Computer Science

Fall 2026

Problem Set 4

Module: Randomization

Below is a reminder of key aspects of the PSet:

- **The only goal of this PSet is to help you develop your problem-solving skills.** Your performance on this PSet will not directly contribute to your grade, but will indirectly improve your ability to do well on the exams.
 - **Your performance does not directly impact your grade**, so you may use any resources you like (collaboration, AI, etc.) to help you complete the PSet.
 - **AI is not equally welcome in every part of this PSet**, and we say so explicitly where it matters. Problem 1 is exam-style and should be attempted alone first. Problem 2 explicitly calls for AI assistance in part (d). Problem 3 is an optional challenge problem that was designed to be solved without AI assistance.
 - **None of this is an integrity rule** – PSets are not graded directly. It's advice about what will actually develop your skills and in the process prepare you well for the exams, which are closed-book and closed-AI.
 - **See the course website** for a full walkthrough of AI tools and recommendations, as well as the input files needed for Problem 2.
 - **An AI-helper is available for Problem 1 and Problem 2(a)–(c)** – unlike a general assistant, it gives hints and pushes back instead of handing you the answer. See the link above for more information.
 - Throughout the PSet, we've included some general tips to help put these into broader context. Exams will not have these, and future PSets may have fewer.
-

Aligning Expectations

The $\blacksquare$ symbol implies that the following problem is an “exam-style” problem. We highly recommend that you write up a full solution to this problem as if you were on an exam.

Problem 1 : Job Processing $\blacksquare$

A distributed system has $n \geq 4$ jobs to complete and n machines, which work in parallel rounds. In each round, every remaining job chooses a machine uniformly at random, independently of the other jobs and of all previous choices. A machine assigned exactly one job completes it; a machine assigned two or more jobs completes none of them. Unfinished jobs try again in the next round. This repeats until all jobs finish. We want to study the number of rounds needed to finish all jobs.

(a) Prove the following three statements. Note that they build on each other, so (i) is needed for (ii), and (ii) is needed for (iii).

(i) Suppose job is still unfinished at the beginning of a round, then it finishes in this round with probability at least $1/3$.

(ii) Let Y_t be the number of jobs left after t rounds, so that $Y_0 = n$. Then, $\mathbb{E}[Y_t] \leq (2/3)^t n$.

(iii) With probability at least $24/25$, at most $n/4$ jobs remain after 12 rounds.

Small Hint

For (i), a job finishes exactly when no other remaining job picks its machine. You may use $(1 - 1/n)^{n-1} \geq 1/e \geq 1/3$ for $n \geq 2$ without proof. (this follows from the $1 + x \leq e^x$ inequality, which we’ve seen a few times now).

For (ii), don’t track Y_t directly. Instead, fix one job and bound the probability that it is still unfinished after t rounds.

For (iii), you need to use a concentration inequality.

(b) Suppose a round starts with b remaining jobs. Prove that the expected number left after this round is at most b^2/n , and that with probability at least $1/2$, at most $2b^2/n$ jobs remain.

(c) Call a round that starts with cn remaining jobs productive if it ends with at most $2c^2n$ jobs. Let

$$K = \lceil \log_2 \log_2 n \rceil.$$

Starting with at most $n/4$ jobs, prove that all jobs finish once K productive rounds have occurred. These rounds need not be consecutive: other rounds may occur between them.

Small Hint

Track the fraction of jobs remaining after each productive round and find a recurrence for it. The recurrence becomes simpler if you look at twice that fraction. Solving the recurrence gives

you the answer.

(d) With K as in (c), prove that all jobs finish within $12 + 4K$ rounds with probability at least $1/4$.

Small Hint

You need to combine the results of (a)–(c). Use (a) to bound the first 12 rounds, then (b) to bound the next $4K$ rounds.

What We Constructed

This analysis shows that a simple randomized rule completes all n jobs in $O(\log \log n)$ rounds with constant probability. At first, collisions are common — meaning that many jobs don't finish in a round — and each round only removes a constant fraction of the jobs in expectation. Once few jobs remain, collisions become rare, and the fraction of remaining jobs roughly squares in every productive round. This is an example of a two-phase analysis, where the first phase is analyzed with one technique and the second phase with another. The two phases are combined to give a bound on the total number of rounds.

Problem 2 : Communication Agreement

A network of sensors has collected temperature readings. We want every sensor to learn the average of these initial readings, using only exchanges between neighbors. Instead of gathering all readings at one place, two neighboring sensors can replace their current values by their average. Repeating this operation may bring the values closer together. Which pairs should communicate when the number of messages is limited?

There are $n \geq 2$ devices, numbered $1, \dots, n$, with initial real values $x_1(0), \dots, x_n(0)$. A connected undirected graph specifies which pairs may communicate: an edge $\{u, v\}$ permits an exchange between devices u and v . They send each other their current values, then both replace their value by $(x_u + x_v)/2$, computed before either value changes. For example, values 10 and 14 both become 12. All other devices keep their values. Exchanges occur one at a time, and each costs two messages, one in each direction.

Write $x(t) = (x_1(t), \dots, x_n(t))$ for the array after t exchanges. The target is the fixed average of the original readings, $\mu = \frac{1}{n} \sum_i x_i(0)$. So the goal is to have $x_i(t) \approx \mu$ for all i after a small number of exchanges. Parts (a)–(c) use exact arithmetic.

(a) Prove the following three statements.

- (i) Exchanges preserve the average, that is, $\frac{1}{n} \sum_i x_i(t) = \mu$ for all t .
- (ii) Suppose $n = 2^r$ (for some positive integer r) and every pair may communicate, in other words, the underlying graph is a complete graph with n vertices. There is a sequence of $(n/2) \log_2 n$ exchanges, costing $n \log_2 n$ messages, after which every device holds exactly μ .

- (iii) In contrast, starting from $(1, 0, 0)$, three devices cannot all have value $\mu = 1/3$ after any finite sequence of exchanges.

- (b) Define the disagreement of an array x by

$$D(x) = \sum_{i=1}^n (x_i - \mu)^2.$$

It is zero exactly when every device holds μ . Prove the following three statements.

- (i) An exchange on $\{u, v\}$ decreases the disagreement by exactly $(x_u - x_v)^2/2$.
(ii) Every array x with average μ satisfies

$$\sum_{u < v} (x_u - x_v)^2 = nD(x),$$

where the sum is over all $\binom{n}{2}$ unordered pairs.

- (iii) Again, assume every pair may communicate, in other words, the underlying graph is a complete graph. Choose one of the $\binom{n}{2}$ unordered pairs uniformly at random and perform an exchange on it. The resulting array x' satisfies

$$\mathbb{E}[D(x') \mid x] = \left(1 - \frac{1}{n-1}\right) D(x).$$

Small Hint

For (i), only two terms in D change during an exchange. For (ii), write $a_i = x_i - \mu$, expand the square, and use $\sum_i a_i = 0$. For (iii), average (i) over all pairs and use (ii).

- (c) Again, assume every pair may communicate, in other words, the underlying graph is a complete graph. Perform T exchanges one after another, each time choosing a pair as in (b)(iii): uniformly at random, and independently of all earlier choices. The arrays $x(1), x(2), \dots$ are now random, since they depend on which pairs were chosen.

- (i) Prove that

$$\mathbb{E}[D(x(T))] = \left(1 - \frac{1}{n-1}\right)^T D(x(0)).$$

- (ii) Let $0 < \varepsilon, \delta < 1$. Show that

$$T = \left\lceil (n-1) \ln \frac{1}{\varepsilon \delta} \right\rceil$$

exchanges suffice to make $D(x(T)) \leq \varepsilon D(x(0))$ with probability at least $1 - \delta$.

- (iii) Putting everything together: suppose every initial reading is within $R \geq 0$ of the average, that is, $|x_i(0) - \mu| \leq R$ for all i , and let $0 < \eta, \delta < 1$. Show that $O(n \log \frac{n}{\eta \delta})$ messages suffice for every device to hold a value within ηR of μ , with probability at least $1 - \delta$. Compare this with the schedule and the result from part (a).

Small Hint

For (i), part (b)(iii) applies whatever the current values are; condition on them, then take expectations and repeat. For (ii), use $1 - z \leq e^{-z}$ and Markov's inequality. For (iii), a single term of D is at most the whole sum.

What We Constructed

Exchanges never change the average, and each one removes a definite amount of disagreement. When every pair can communicate, a random exchange shrinks the disagreement by a factor $1 - 1/(n - 1)$ in expectation. For fixed relative accuracy and failure probability, $O(n \log n)$ messages suffice to bring every device close to the average. This matches the order of the exact schedule in (a) when n is a power of two, while working for every n without a predetermined schedule. Part (d) asks what changes when only some pairs can communicate.

(d) We will now experiment with the same problem on graphs that are not complete. The course website will provide six input graphs with initial readings and varying numbers of vertices and edges. Every input has $D(x(0)) > 0$. Each `.txt` file starts with n and m , followed by a line of n initial readings in vertex order $0, \dots, n - 1$, then m lines listing the endpoints of the undirected edges.

Compare uniformly random edge selection with one alternative you design. It may be randomized or deterministic and may use the graph and previous exchange choices. Have an AI assistant implement the methods and a simulator of the exchange process. Check the implementation by verifying conservation of the average and the one-exchange identity in (b).

Let $E(t) = \max_i |x_i(t) - \mu|$ be the largest error among the devices after t exchanges. Track both $E(t)$ and the disagreement $D(x(t))$ from parts (a)–(c). Plot $E(t)/E(0)$ and $D(x(t))/D(x(0))$ against messages used, repeating randomized methods 10 times to show variability. For each run, report the number of messages until $E(t) \leq 10^{-2}E(0)$ for the first time: every device's error relative to μ must be at most 1% of the initial maximum error.

Compare how the two measures decrease and whether they give the same picture of which method works better, keeping in mind that D measures squared errors while E does not. Explain your design and where it helped or failed. Discuss why the complete-graph guarantee in (c) does not directly apply to these graphs. You need not beat the baseline everywhere.

Aligning Expectations – Using AI in This Question

Part (d) asks you to use AI assistance for implementation and experiments. Do not try to solve this part without using AI assistance. Decide what you want to implement and measure, then ask the assistant to help carry out those plans. You are responsible for checking that the code implements your methods.

Problem 3 : Learning to Learn

Optional – Challenge Problem

This problem is optional and considerably harder than Problems 1 and 2 – we don't suggest attempting it just to practice for the exam. It's here for those who want to engage more deeply with the material; if you're considering an IW/thesis in CS theory, it'll give you a taste of what that might be like. Feel free to discuss it with your TA/UCA coach.

Do not use AI to solve this problem. This is a challenge problem, so if you want to work on it you should do so on your own.

Consider a probability distribution over the set $\{1, 2, \dots, n\}$ which is described by a vector $p = (p_1, p_2, \dots, p_n)$, where p_i is the probability of sampling the element i . This probability distribution is unknown to you and your goal is to find an approximation $\hat{p} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n)$ such that the two distributions are close in the following metric: $\sum_{i=1}^n |p_i - \hat{p}_i| \leq \epsilon^1$.

(a) Show that if p and $\hat{p}$ are close according to the metric above, then for any function $f : \{1, 2, \dots, n\} \rightarrow [0, 1]$, the expected values of f under the two distributions are also close, i.e.,

$$|\mathbb{E}_{i \sim p}[f(i)] - \mathbb{E}_{i \sim \hat{p}}[f(i)]| \leq \epsilon.$$

Note that the above is one way of justifying why we care about approximating the distribution p in this metric.

(b) Suppose you independently sample m times from the p distribution, obtaining samples $x_1, x_2, \dots, x_m$. Let $\hat{p}$ be the empirical distribution, where $\hat{p}_i = \frac{|\{x_j = i\}|}{m}$ is the fraction of samples that equals to i .

Show that if $m \geq \Omega\left(\frac{n^2 \ln(n/\delta)}{\epsilon^2}\right)$, then with probability at least $1 - \delta$, the empirical distribution $\hat{p}$ satisfies

$$\sum_{1 \leq i \leq n} |p_i - \hat{p}_i| \leq \epsilon.$$

¹This metric is very close to a common metric called total variation distance, which is defined as $\frac{1}{2} \sum_{i=1}^n |p_i - \hat{p}_i|$. The factor of $1/2$ is included for technical reasons, for example, it ensures that the distance lies between 0 and 1.