

CS 473 (Fall 2026)

Homework 5 (due Oct 15 Thu 10am)

Instructions: Carefully read <http://courses.grainger.illinois.edu/cs473/fa2026/policies.html#hw> and <http://courses.grainger.illinois.edu/cs473/fa2026/integrity.html> (in particular, remember to include the 3 disclosure statements, on collaborators, outside sources, and AI use, for each problem).

Problem 5.1: (a) (15 pts) Given a set S of n numbers and a subset $R = \{x_1, \dots, x_r\}$ where each element x_i is a random, independently chosen element from S , compute the probability that $\min(R)$ has rank more than k (i.e., there are at least k elements in S with value smaller than $\min(R)$).

(b) (15 pts) Show that $\min(R)$ has rank at most $c(n/r) \log n$ with probability at least $1 - 1/n$, for some constant c . [Note: the inequality $1 - x \leq e^{-x}$ may be useful...]

(c) (70 pts) You are given an undirected graph $G = (V, E)$ stored in the standard adjacency-lists representation. Assume that each vertex has degree at most a constant, say 3. Assume each vertex v holds a real-valued weight $w(v)$. The problem is to find a *local minimum* u , i.e., a vertex u satisfying the property that $w(u) \leq w(v)$ for all vertices v adjacent to u . (There could be more than one local minimum.)

Surprisingly, with randomization it is possible to solve this problem without having to examine all n vertices of the graph! Give a Las Vegas algorithm that finds a local minimum in $O(\sqrt{n \log n})$ time with probability at least $1 - 1/n$.

[Hint: use random sampling. As you would expect, part (b) should come in handy...]

Problem 5.2: Consider the following Monte Carlo algorithm to sort a given array $A[1, \dots, n]$, for a parameter d :

1. repeat d times {
2. i = random number in $\{1, \dots, n\}$
3. j = random number in $\{1, \dots, n\}$
4. if $i < j$ and $A[i] > A[j]$ then
5. swap $A[i]$ and $A[j]$

(a) (25 pts) Let Φ be the current number of pairs (k, ℓ) with $k < \ell$ and $A[k] > A[\ell]$. Show that if line 4 holds and line 5 is executed, then Φ decreases by at least 1.

(b) (25 pts) Fix a moment in time during the execution of the algorithm. Suppose that $\Phi = k$ currently. Show that the expected number of iterations in the repeat loop till the next swap (line 5) is executed is $O(n^2/k)$.

(c) (50 pts) Describe how to set d (aim for the smallest bound) to guarantee that the algorithm solves the sorting problem with error probability at most 0.001. [Hint: use (b). Aside: would you recommend this algorithm to practitioners?]

Optional Problem 5.3 (ungraded): In this problem, we will explore other simple families of hash functions and see how close they are to being universal. Let $U \geq m$.

- (a) (35 pts) Let p be a random prime in $[m/2, m]$. Let b be a random number in $\{0, \dots, p-1\}$. Define $h_p : \{0, \dots, U-1\} \rightarrow \{0, \dots, m-1\}$ by

$$h_p(x) = x \bmod p.$$

Prove that for any fixed $x, y \in \{0, \dots, U-1\}$ with $x \neq y$, we have $\Pr_p[h_p(x) = h_p(y)] \leq O((\log U)/m)$.

(Thus, this hash function family is “almost” universal with an extra logarithmic factor.)

[Hint: By the prime number theorem, the number of primes in $[m/2, m]$ is $\Theta(m/\log m)$. Given a number $z \leq U$, how many prime factors in $[m/2, m]$ can z have?]

- (b) (35 pts) Suppose $U = 2^w$ and $m = 2^\ell$. We can view numbers in $\{0, \dots, U-1\}$ as binary vectors in $\{0, 1\}^w$, and numbers in $\{0, \dots, m-1\}$ as binary vectors in $\{0, 1\}^\ell$.

Pick a random 0-1 matrix A of dimension $\ell \times w$ (each of its ℓw entries is an independently chosen random bit). Define $h_A : \{0, 1\}^w \rightarrow \{0, 1\}^\ell$ by

$$h_A(x) = (Ax) \bmod 2.$$

All arithmetic operations are done modulo 2; for example, addition is equivalent to exclusive-or.

Prove that for any fixed $x, y \in \{0, 1\}^w$ with $x \neq y$, we have $\Pr_A[h_A(x) = h_A(y)] = 1/m$.

(Thus, this hash function family is universal, and its computation avoids integer division and requires only bitwise exclusive-or, which is cheaper. However, the evaluation time isn't $O(1)$.)

[Hint: suppose x and y differs at the i -th bit. Focus on choices over the i -th column of A ...]

- (c) (30 pts) Continuing part (b), let b be a random 0-1 vector of dimension ℓ . Define $h_{A,b} : \{0, 1\}^w \rightarrow \{0, 1\}^\ell$ by

$$h_{A,b}(x) = (Ax + b) \bmod 2.$$

Prove that for any fixed $x, y \in \{0, 1\}^w$ with $x \neq y$ and for any fixed $s, t \in \{0, 1\}^\ell$, we have $\Pr_{A,b}[(h_{A,b}(x) = s) \wedge (h_{A,b}(y) = t)] = 1/m^2$.

(Thus, this hash function family is 2-universal.)

[Hint: use (b).]