**Mathematical Toolkit** **Fall 2026**

# Homework 3

Due: November 4, 2026

**Note**: *You may discuss these problems in groups. However, do not look up solutions online or "look up" solutions using AI. Also, you must write up your own solutions and mention the names of the people in your group. Also, list any books, papers or other sources you refer to. It is recommended that you typeset your solutions in LaTeX. Homeworks are due by 11:59PM on the due date.*

1. **Random Orderings.**
   Let $\Omega$ be the set of all $n!$ permutations of $\{1, ..., n\}$, and $\nu$ be the uniform distribution over $\Omega$. That is, $\nu(\omega) = 1/n!$ for each permutation $\omega \in \Omega$.

   (a) Given permutation $\omega = (\omega_1, \ldots, \omega_n)$, an *inversion* is a pair $(i, j)$ such that $i < j$ but $\omega_i > \omega_j$. Let $X(\omega)$ be the number of inversions in $\omega$. What is $\mathbb{E}[X]$? Be clear about your reasoning.

   (b) We say that position $i$ is a *prefix-maximum* in permutation $\omega = (\omega_1, \ldots, \omega_n)$ if $\omega_i > \omega_j$ for all $j < i$. Let $Y(\omega)$ be the number of prefix maxima in $\omega$. What is $\mathbb{E}[Y]$? Explain your reasoning.

2. ***t*-Universal Hashing**.
   We say that $H$ is *t-universal over range M* if, for every fixed sequence of $t$ distinct inputs $s_1, s_2, \ldots, s_t$ and for any $h$ chosen at random according to $H$, the sequence $h(s_1), h(s_2), \ldots, h(s_t)$ is equally likely to be any of the $M^t$ sequences of length $t$ with elements drawn from $\{0, 1, \ldots, M-1\}$. It's easy to see that if $H$ is 2-universal then it is universal.

   Consider strings $s$ of length $n$ from the alphabet $\{0, 1, \ldots, k-1\}$ where $k \geq 2$ . One way to construct a hash function over such inputs is to generate a 2-dimensional $n \times k$ table $T$ of $b$-bit random numbers where $b = \lg(M)$. The first index of $T_{i,j}$ is in the range $[1, n]$ and the second index is in the range $[0, k-1]$. Here the hash function $h_T$ is defined as follows:

   $$h_T(s) = \bigoplus_{i=1}^{n} T_{i,s[i]}$$

   where $s[i]$ is the $i$th character in $s$ and "$\bigoplus$" represents the xor function of integers represented in binary.

   (a) Prove that this construction is *not* 4-universal.

(b) Prove that this construction *is* 3-universal.

3. **MAX-SAT Revisited.**
In class, we saw that if $F$ is a $k$-CNF formula of $m$ clauses in which every clause has size exactly $k$ (and you are not allowed to repeat variables inside a clause), then there must exist an assignment satisfying at least $\lceil m(1 - 1/2^k) \rceil$ clauses of $F$. In this question, you will give an efficient deterministic algorithm to find such an assignment.

   (a) Prove the following claim: Suppose we have a CNF formula $F$ of $m$ clauses, with $m_1$ clauses of size 1, $m_2$ of size 2, etc. ($m = m_1 + m_2 + ...$). Let $X$ be a random variable denoting the number of clauses satisfied in a random assignment. Then $\mathbb{E}[X] = \sum_k m_k(1 - 1/2^k)$.

   (b) Here is a deterministic algorithm. Compute $E = \mathbb{E}[X]$ using the formula from part (a), and then compute $E_0 = \mathbb{E}[X|x_1 = 0]$ and $E_1 = \mathbb{E}[X|x_1 = 1]$ using a similar formula. Set $x_1$ based on whichever of $E_0$ and $E_1$ is larger. Now, recurse on the formula remaining.

      i. Explain how you can efficiently compute $E_0$ and $E_1$ in this algorithm.

      ii. Explain why this algorithm is guaranteed to find a solution that satisfies at least $E = \mathbb{E}[X]$ clauses.

   Note: this approach is called the "conditional expectation method"