
HW I: Basic probability (due Feb 7th, before tutorials)

1. In class, we covered a randomized algorithm to check the product of matrices over \mathbb{F}_2 . In an analogous way, design and analyze a randomized algorithm for checking the product of matrices over \mathbb{F}_p where p is a prime number. The error probability should be at most $1/100$.

2. Show that the events $\{A_i\}_{1 \leq i \leq n}$ are mutually independent if and only if

$$\mathbf{P} \left\{ \bigcap_{i=1}^n B_i \right\} = \prod_{i=1}^n \mathbf{P} \{B_i\}$$

where for every i , either $B_i = A_i$ or $B_i = A_i^c$. We use the notation A^c for the complement of A in Ω .

3. Write an algorithm that takes as input n and generates a uniformly random permutation of $\{1, \dots, n\}$ (represented in an array in the natural way). You can use the function $\text{RandInt}(m)$ which returns a uniform number between $\{1, \dots, m\}$. You should justify why each permutation has probability $\frac{1}{n!}$. You should aim for a running time of $O(n)$, where you may assume that a call to RandInt takes constant time and that accessing a given index of an array takes constant time. Less efficient (but correct) algorithms will be given partial credit.