

1 Inversion

Let A be an array of size n containing n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion in A . Suppose that each element of A is chosen uniformly at random from the range 1 through n . Use indicator random variables to compute the expected number of inversions.

Let X_X represent the indicator random variable when an inversion in the pair (i, j) occurs in array A . For $1 < i \leq j \leq n$, let $P(X_X = 1) = 1/2$. This implies that given two random numbers from the array A , assigned to i and j , there is a 50% chance that an inversion will occur. Therefore, $E[X_X] = \frac{1}{2}$. Next, I let the X represent all of the inversions that occur in A such that

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^n X_X$$

and finally, the expected number of pairs of inversions to occur in A is represented by

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^n X_X\right]$$

Using the linearity of expectations, substituting in $E[X_X]$ stated above and the fact that the two summations can be represented as $\frac{n(n-1)}{2}$ as that is how many pairs there such that an inversion could occur in A , the following can be done to get the expected number of inversions in A :

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^n X_X\right] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n E[X_X] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{2} \\ &= \frac{n(n-1)}{2} * \frac{1}{2} \\ &= \frac{n(n-1)}{4} \end{aligned}$$

2 Lottery numbers

Some lotteries award enormous prizes to people who correctly choose a set of seven numbers from 1 to 60. What is the probability of winning such a lottery.

The combination, $\binom{60}{7}$ represents the number of ways that the seven lottery numbers can be picked. 1 Divided by said combination will be the probability of the event that a person will pick the correct numbers,

$$\frac{1}{\binom{60}{7}} = \frac{1}{386206920} \approx 2.6 * 10^{-9}$$

Essentially, there is an extremely miniscule chance that someone is able to pick the correct lottery numbers.

3 Bit Sequence

A sequence of 16 bits is generated uniformly at random. What is the probability that at least one of these bits is 0?

A bit has 2 states, 1 or 0, each having the same chance to occur, $\frac{1}{2}$. The complement of an event is the probability that said event won't occur and is represented by $(1 - p)$. The complement of the event that there is at least one bit that is 0 out of the 16 is the event that all 16 bits are 1. Because the state of the bits are chosen uniformly random and are independent of each other, the probability of all 16 bits having a state of 1 is $(\frac{1}{2})^{16}$. Therefore, the probability that at least one of the 16 bits is 0 is,

$$(1 - (\frac{1}{2})^{16}) \approx 0.99998$$

This means that is nearly guaranteed that at least one of the 16 bits will be a 0.

4 Unbiased-Random

Suppose that you want to write a routine that will return a 0 with probability 1/2 and a 1 with probability 1/2. At your disposal is a procedure called `Biased_Random`, that returns either a 0 or 1. It returns 0 with some probability p and returns 1 with probability $1 - p$, where $0 < p < 1$. Unfortunately you do not know what p is. Give an algorithm that uses `Biased_Random` as a subroutine, and returns an unbiased answer. Your algorithm should return 0 with probability 1/2 and return 1 with a probability 1/2. Explain why your algorithm is correct.

My `unbiased_random` algorithm would call `biased_random` twice and compare the what was returned. If the two values returned from `biased_random` are the same value, `biased_random` is called twice again and this will continue until the two values returned are

different. When this happens, my unbiased_random algorithm would return the first value returned from biased_random between the two calls to that algorithm.

To prove that my algorithm returns an unbiased answer, let the first call of biased_random be represented by a and the second call represented by b . The probabilities of these variables being different is the following:

$$\begin{aligned}P(a \neq b \text{ and } b = 1) &= (1 - p) * p \\P(a \neq b \text{ and } b = 0) &= p * (1 - p)\end{aligned}$$

The values $(1 - p) * p$ and $p * (1 - p)$ are equivalent and therefore each have an equal chance of occurring proving that my algorithm would return an unbiased random number by onling calling biased_random.