1 Probability

Let $X$ be a random variable which is 1 with probability $\frac{1}{100}$ and 0 with probability $\frac{99}{100}$.

1. What is the expected value $E[X]$?

2. Suppose you draw $n$ independent random variables, $X_1, X_2, \ldots, X_n$, distributed like $X$. What is the expected value $E[\sum_{i=1}^{n} X_i]$?

3. Suppose I draw independent random variables $X_1, X_2, \ldots$ and I stop when I see the first “1”. For example, if I draw

   $$X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 1$$

then I would stop at $X_4$. Let $N$ be the last index that we draw. (So in the previous example, $N = 4$). How big do you expect $N$ to be?
2 Duplicate Ducks

There are \( n \) ducks standing in a line. Every duck has a unique height except \( k \) ducks that are identical and have the same height (they were born together!!). Knowing that, the duplicate ducks' mother is wondering how tall have they grown, and she wants to find their heights' value. She has not taken CS161 yet, but she is really into randomized algorithms, and she has come up with 2 different randomized algorithm (along with a helper function) to find her \( k \) identical children's heights.

For each of the two randomized algorithms, answer the following questions:

1. Is this a Las Vegas or Monte Carlo randomized algorithm?
2. What is the worst case runtime of the randomized algorithm?
3. What is the expected runtime of the randomized algorithm?
4. What is the probability of returning a correct output?

Here are the algorithms:

Algorithm 1 findDuplicateDuck1
Input: An array \( D \) of duck heights, and \( k \)
while true do
\[
\text{Choose a random } d \in D \\
\text{if hasKDuplicates}(D, k, d) \text{ then} \\
\quad \text{return } d
\]

Algorithm 2 findDuplicateDuck2
Input: An array \( D \) of duck heights, and \( k \)
for 10 iterations do
\[
\text{Put the ducks } D \text{ in random order, using } O(n) \text{ time} \\
\quad d \leftarrow D[0] \\
\quad \text{if hasKDuplicates}(d) \text{ then} \\
\quad \quad \text{return } d
\]
return \( D[0] \)

Algorithm 3 hasKDuplicates
Input: An array \( D \) of duck heights, \( k \), and a duck height \( d \)
Output: True if there are \( k \) ducks in \( D \) that also have height \( d \)
count \( \leftarrow 0 \)
for \( x \in D \) do
\[
\text{if } x == d \text{ then} \\
\quad \text{count} \leftarrow \text{count} + 1
\]
return \( \text{count} == k \)

Figure 1: Two randomized algorithms for finding the height of the duplicate ducks
3 Light Bulbs and Sockets

You are given a collection of $n$ differently sized light bulbs that have to be fit into $n$ sockets in a dark room. You are guaranteed that there is exactly one appropriately-sized socket for each light bulb and vice versa; however, there is no way to compare two bulbs together or two sockets together as you are in the dark and can barely see! (You are, however, able to see where the sockets and light bulbs are.) You can try and fit a light bulb into a chosen socket, from which you can determine whether the light bulb’s base is too large, too small, or is an exact fit for the socket. If the bulb fits exactly, it will flash once, in which case you have a correct match. (Note that the flashing light does not allow you to visually compare bulbs/sockets to other bulbs/sockets.)

Suggest a (possibly randomized) algorithm to match each light bulb to its matching socket. Your algorithm should run strictly faster than quadratic time in expectation. Give an upper bound on the worst-case runtime, then prove your algorithm’s correctness and expected runtime.