# CS 281: Challenge #8

*Professor Alexander Sherstov*

**Nakul Khambhati**

# Problem 20

For this question, I have a good idea of how to simulate a fair coin however, I'm not sure how to evaluate the expected running time as my description of the method is somewhat vague. We are given a coin that outputs 1 with probability $p$ so it outputs 0 with probability $1-p$. We can flip the coin a few times (this is where I'm not sure how many flips are needed) and list all the outcomes in succession. Let's say after 10 trials, we get 1001011011. (It is reasonable to expect that $p = 0.6$ here.) We can simulate a fair coin as follows:

1. Pair up the outcomes as follows: 10 01 01 10 11.
2. Notice that the pair 11 has probability $p^2$ of occuring and 00 has probability $(1-p)^2$ however both 10 and 01 have probability $p(1-p)$.
3. We can then delete all pairs 11 and 00 from the list of outcomes and upon counting, since 01 and 10 have equal probability of occuring, we should get an equal number of these patterns.

Thus, we have proved that we can simulate a fair coin by flipping a biased coin a lot of times, pairing outcomes, deleting 11 and 00, and setting heads = 10, tails = 01.

# Problem 21

We want to be able to create an event with arbitrary probability which we write as $p = 0.b_1b_2b_3\cdots$. We are only allowed to use a fair coin to simulate this event. First, I will provide some motivation for my method. It is easy to simulate an event that occurs with probability $\frac{1}{2^k}$. For example, flipping a coin twice, the outcome 11 has probability $\frac{1}{4}$. The expected number of flips here to obtain a positive outcome is 2. However, we can short-circuit the flipping process in some cases to reduce this value. For example, if the first flip reveals 0 then it is impossible to get 11 with the next flip so we can abort the process. This brings down the expected number of flips to $\frac{3}{2}$.

Now, I will explain my method for simulating an event with arbitrary probability. Let $p = 0.b_2b_2b_3\cdots$. Flip a coin until it shows 1. Let's say it takes $n$ flips for the first 1. Output $b_n$. In words, output the bit indexed by the number of flips it takes for the first positive outcome. It may not yet be clear that the probability of outputting 1 here is exactly $p$.

To see this, note that the probability that the process stops after $n$ flips is exactly $(\frac{1}{2})^n$ so the probability of outputting 1 is

$$
\begin{aligned}
&\mathbb{P}[(n=1 \wedge b_1 = 1) \vee (n=2 \wedge b_2 = 1) \vee (n=3 \wedge b_3 = 1) \vee \cdots] \\
&= \frac{1}{2}b_1 + \frac{1}{2^2}b_2 + \frac{1}{2^k}b_3 + \cdots \\
&= \sum_{i=1}^{\infty} \frac{1}{2^i}b_i \quad \text{(binary expansion of } p\text{)} \\
&= p
\end{aligned}
$$

In this way, we have constructed (using only a fair coin) an event that succeeds with probability $p$ for any $p \in (0,1)$. Moreover, we expect to see a positive outcome after 2 flips so the expected runtime of this is $O(1)$.