# CSE 312: Foundations of Computing II Summer 2022

## Problem Set 4 (due Friday, July 22, 11:59pm)

**Directions**: *For each problem, explain/justify how you obtained your answer, as correct answers without an explanation may receive **no credit**. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide. Unless you are asked to, you should leave your answer in terms of factorials, combinations, etc., for instance $26^7$ or $26!/7!$ or $26 \cdot \binom{26}{7}$.*

***Submission***: *You must upload a **pdf** of your solutions to Gradescope under "Pset 4 [Written]". The use of LaTex is highly recommended, and we have provided a template. (Note that if you want to hand-write your solutions, you'll need to scan them. If we cannot make out your writing, your work may be ungradable, so make sure it is legible.) Your code will be submitted as a .py file under "Pset 4 [Coding]".*

*Instructions as to how to upload your solutions to Gradescope are on the course web page.*

*Remember that you must tag your written problems on Gradescope, or you will potentially receive **no credit** as mentioned in the syllabus. Please put each numbered problem on its own page in the pdf (this will make selecting pages easier when you submit), and ensure that your pdfs are oriented correctly (e.g. not upside-down or sideways). As stated above, the coding problem will also be submitted to Gradescope.*

***Collaboration***: *This pset must be submitted **individually**. You are welcome and encouraged to discuss approaches with your fellow students, but everyone **must write up their own solutions.** Failure to do so is an instance of academic dishonesty.*

## 1. Busy 312 Students (10 points)

CSE 312 students sometimes delay laundry for a few days (to the chagrin of their roommates).

A ***busy*** 312 student must complete 3 problem sets before doing laundry. Each problem set requires 1 day with probability 2/3 and 2 days with probability 1/3. (The time it takes to complete different problem sets is independent.) Let $B$ be the number of days a busy student delays laundry. What is the probability mass function for $B$?

## 2. Doggies and Koalas (25 points)

Suppose that $4n$ animals are partitioned into pairs at random, with each partition being equally likely. If the set consists of $n$ doggies and $3n$ koalas, what is the expected number of doggie-koala couples?

## 3. More Coin Flipping (25 points)

A coin with probability $p$ of coming up heads is tossed independently $n$ times. What is the expected number of "maximal runs" (which is a maximal sequence of consecutive flips that are the same)? For example, the sequence HHHTTHTHHH has 5 runs, the first three H, the following two T, and so on. Use linearity of expectation, carefully define indicator rvs, and justify your work.

## 4. Packet Failures (20 points)

Consider three different models for sending $n$ packets over the Internet:

a. each packet takes a different path. Each path fails independently with probability $p$;

b. all packets take the exact same path which fails with probability $p$. Thus, either all the packets get through or none get through;

c. half the packets take one path, and half take the other (assume $n$ is even), and each of the two paths fails independently with probability $p$.

Let $A, B,$and $C$ be RVs representing the number of packets lost for cases $a, b,$and $c$ respectively. Write down the probability mass function, the expectation and the variance for each.

## 5. Bloom Filter [Coding] (20 points)

Google Chrome has a huge database of malicious URLs, but it takes a long time to do a database lookup (think of this as a typical `Set`). They want to have a quick check in the web browser itself, so a space-efficient data structure must be used. A **bloom filter** is a **probabilistic data structure** which only supports the following two operations:

I. `add(x): Add an element` $x$ `to the structure.`

II. `contains(x): Check if an element` $x$ `is in the structure. If either returns "definitely not in the set" or "could be in the set".`

It does **not** support the following two operations:

I. Delete an element from the structure.

II. Give a collection of elements that are in the structure.

The idea is that we can check our bloom filter if a URL is in the set. The bloom filter is always correct in saying a URL definitely isn't in the set, but may have false positives (it may say a URL is in the set when it isn't). Only in these rare cases does Chrome have to perform an expensive database lookup to know for sure.
Suppose we have $k$ **bit arrays** $t_1, \ldots, t_k$ each of length $m$ (all entries are 0 or 1), so the total space required is only $km$ bits or $km/8$ bytes (as a byte is 8 bits). Suppose the universe of URL's is the set $\mathcal{U}$ (think of this as all strings with less than 100 characters), and we have $k$ ***independent and uniform*** hash functions $h_1, \ldots, h_k : \mathcal{U} \rightarrow \{0, 1, \ldots, m-1\}$. That is, for an element $x$ and hash function $h_i$, pretend $h_i(x)$ is a **discrete** $Unif(0, m-1)$ random variable. Suppose we implement the `add` and `contains` functions as follows:

**Bloom Filter Operations**

```
1: function INITIALIZE(k,m)
2:     for $i = 1, \ldots, k$: do
3:         $t_i$ = new bit array of m 0's
4: function ADD(x)
5:     for $i = 1, \ldots, k$: do
6:         $t_i[h_i(x)] = 1$
7: function CONTAINS(x)
       return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$
```

Refer to the notes from Chapter 9.4 in the textbook for more detail on bloom filters. Implement the functions `add` and `contains` in the `BloomFilter` class of `cse312_pset4_bloom.py`.