L22

1. You have a coin that turns up heads with probability P. How many tosses do you need in expectation before you encounter the pattern HT consecutively (including the tosses HT themselves)?

Let E[X] be the expected number of coin tosses before landing on T.

Toss	Probability
H	$p(1 + E[X])$
T	$1 - p$

If the first toss results in heads, additional tosses are required until tails is obtained.

If the first toss lands on tails, no further tosses are needed.

$$
\begin{aligned}
E[X] &= (1-p) + p(1 + E[X]) \\
E[X] &= 1 - p + p + p \cdot E[X] \\
E[X] &= 1 + p \cdot E[X] \\
E[X](1-p) &= 1 \\
E[X] &= \frac{1}{1-p}
\end{aligned}
$$

Let E[Y] be the expected number of coin flips until we get a HT sequence

Toss	Probability
H	$(1 + E[X])$
T	$(1 - p)(1 + E[Y])$

If the initial coin flip results in tails, we'll need to restart with an additional flip.

$$
\begin{aligned}
E[Y] &= (1-p)(1+E[Y]) + p(1+E[X]) \\
E[Y] &= (1-p) + E[Y] - pE[Y] + p + pE[X] \\
E[Y] &= 1 + E[Y] - pE[Y] + pE[X] \\
pE[Y] &= 1 + pE[X] \\
pE[Y] &= 1 + \frac{p}{1+p} \\
E[Y] &= \frac{1}{p} + \frac{1}{1-p}
\end{aligned}
$$

2. A fancy search algorithm is shown below. Suppose a key k is present in the array a, which contains unique numbers. What is the running time of this algorithm in expectation?

```
def fancy_search(a: List[int], k: int) -> int:
    random_loc = pick a number between 0 and len(a) - 1 unifo
    while a[random_loc] != k:
        random_loc = pick a number between 0 and len(a) - 1 u
    return random_loc
```

Let T(n) denote the sum of all possible sizes obtained from selecting a pivot, where each size is multiplied by the probability associated with selecting that particular pivot. This calculation encompasses the time needed to select another random pivot. Given that pivot selection is random, the probability of selecting any specific pivot is p = $\frac{1}{n}$

$$
\begin{aligned}
T(n) &= T(1) + T(2) + T(3) + ... + T(n-1) + 1 \\
nT(n) &= T(1) + T(2) + T(3) + ... + T(n-1) + n \qquad (1) \\
(n-1)T(n-1) &= T(1) + T(2) + T(3) + ... + T(n-2) + n - 1 \qquad (2)
\end{aligned}
$$

(1) - (2):

$$
\begin{aligned}
nT(n) - (n-1)T(n-1) &= T(n-1) + 1 \\
nT(n) &= nT(n-1) + 1
\end{aligned}
$$

Dividing both sides by n, we have the recurrence:

$$T(n) = T(n-1) + \frac{1}{n}$$

We can solve this recurrence by unfolding it:

$$\begin{aligned} T(n) &= T(n-1) + \frac{1}{n} \\ &= T(n-2) + \frac{1}{n-1} + \frac{1}{n} \\ &= T(n-3) + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n} \\ &= 1 + ... + \frac{1}{n-3} + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n} \\ T(n) &= \sum_{i=1}^{n} \frac{1}{n} \end{aligned}$$

his is a harmonic series, so therefore $T(n) = O(log(n))$.