A PUZZLE OF TOSSING COINS

UMESH P. NARENDRAN

1. Question

A large number of people are tossing unbiased coins that have equal probability for heads and tails. Each of them tosses a coin until he/she gets a tail. So, the tosses by different people may be like this:

```
T
HT
HHT
HHHT
HHHHT
...
```

If a large number of people are doing this, what is the average number of tosses (i.e., total coin tosses divided by number of people) a person makes?

2. Answer

2 tosses, on average.

3. Solution

3.1. **Approach 1.** Irrespective of all other conditions, heads and tails occur with equal probability, so the number of heads will be equal to the number of tails ultimately.

Now, for n people, the number of total tails is exactly equal to n, because every person continues tossing until he/she gets exactly one tail.

Since the number of heads and tails are approximately equal, there will be approximately n heads. So, there will be a total of approximately $2n$ tosses.

This means a person does approximately 2 tosses on average.

3.2. **Approach 2.** Let there be n people.

In the first round, approximately $\frac{n}{2}$ heads and $\frac{n}{2}$ tails result. People who got tails will stop there.

In the second round, of the $\frac{n}{2}$ tosses, approximately $\frac{n}{4}$ will give heads and $\frac{n}{4}$ will give tails.

In the third round, of the $\frac{n}{4}$ tosses, approximately $\frac{n}{8}$ will give heads and $\frac{n}{8}$ will give tails.

So, total number of tosses $= \left(\frac{n}{2}+\frac{n}{2}\right)+\left(\frac{n}{4}+\frac{n}{4}\right)+\left(\frac{n}{8}+\frac{n}{8}\right)+\cdots = n\left(1+\frac{1}{2}+\frac{1}{4}+\cdots\right) = n\cdot\frac{1}{1-\frac{1}{2}} = 2n.$

So, average number of tosses per person $= \frac{2n}{n} = 2.$

3.3. **Approach 3.** There is a probability of $\frac{1}{2}$ to get a tail in one toss, $\frac{1}{4}$ chance to get in 2 tosses, $\frac{1}{8}$ chance to get in 3 cases etc.

So, the average number of tosses is

$$1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + \cdots = \sum_{k=1}^{\infty} \frac{k}{2^k}$$

There are many ways to find this sum. Given below are two of them.

3.3.1. *Representing as the sum of two serieses.* Let

$$\begin{aligned} x &= \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \cdots \\ &= \frac{1}{2} + \left(\frac{1}{4} + \frac{1}{4}\right) + \left(\frac{1}{8} + \frac{2}{8}\right) + \left(\frac{1}{16} + \frac{3}{16}\right) + \cdots \\ &= \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots\right) + \left(\frac{1}{4} + \frac{2}{8} + \frac{3}{16} + \cdots\right) \\ &= \frac{\frac{1}{2}}{1 - \frac{1}{2}} + \frac{x}{2} \end{aligned}$$

So, $x = 2 \cdot \frac{\frac{1}{2}}{1-\frac{1}{2}} = 2$.

3.3.2. *Deriving by another method.* When $-1 < x < 1$,

$$1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}$$

Differentiating,

$$0 + 1 + 2x + 3x^2 + 4x^3 + \cdots = \frac{1}{(1-x)^2}$$

Multiply by x,

$$x + 2x^2 + 3x^3 + 4x^4 + \cdots = \frac{x}{(1-x)^2}$$

Putting $x = \frac{1}{2}$,

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \cdots = \frac{\frac{1}{2}}{\left(\frac{1}{2}\right)^2} = 2$$

4. Further questions

It may come as a surprise that the average number of tosses if only two. In fact, there will be many tosses for many people, but they will be considerably less in number. Also, the maximumm number of people (around 50%) have only one toss. These make average value to 2.

It may be of interest to check the range of number of tosses when n people are involved. Since around half of the people stop at each toss, it is easy to see that it will take approximate $\lceil \log_2 n \rceil$ tosses so all the people will get a tail.

5. Simulation

Simulation is the best way to verify a probability hypothesis. When we try with bigger samples, we get answers closer to the theoretical result. It happens here as well.

5.1. **The program.** I wrote the following Java program to simulate this.

```
// ------------------------------------------------------------------------
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Random;

// Simulates coin tosses till the occurrence of a Tail.
// Collects statistics to check whether the theoretical probability is accurate.
 public class TillTailToss {

  // Handles the toss done by a person.
   private class Person {
    // To make sure the tosses are random and independent, each person is
    // given a random number generator, rather than using a common one.
     private Random random;

    // Indicates whether this person has got a Tail.
     private boolean done;

    // Total number of tosses, till the tail.
     private int nTosses;

    // Constructor for TillTailToss.Person.
     public Person() {
      random = new Random();
      done = false;
      nTosses = 0;
    }

    // Do the next toss.  Collect the statistics.  If it is a tail,
    //  make sure there will not be any other tosses.
     public boolean tossNext() {
       if (done) {
        return true;
      }
      ++nTosses;
       if (random.nextInt(2) == 0) {
        done = true;
      }
       return done;
    }
```

```
  // Return the number of tosses.
   public int tosses() {
    return nTosses;
   }
}

 private List<Person> people = null;

// Constructor for TillTailToss.
// The parameter specifies how many people should be simulated.
 public TillTailToss(int n) {
  people = new ArrayList<Person>();
   for ( int i = 0; i < n; ++i) {
    people.add(new Person());
   }
}

// Do the next round of tosses with all the people involved.
// Returns true if all of them got tail, false otherwise.
 public boolean tossNext() {
   boolean done = true;
   for (Person person : people) {
     if (person.tossNext() == false) {
       done = false;
     }
   }
   return done;
}

// Do tosses until everyone gets a tail.
// Returns the number of rounds of tosses.
 public int doTosses() {
   int rounds = 0;
   do {
    ++rounds;
   }  while (!tossNext());
   return rounds;
}

// Reports the statistics collected.
 public void report() {
  // Constructs a frequency table.
  Map<Integer, Integer> tossCounts =  new HashMap<Integer, Integer>();
   for (Person person : people) {
    Integer tosses = person.tosses();
    Integer freq = tossCounts.get(tosses);
    tossCounts.put(tosses, (freq == null ? 1 : freq + 1));
```

```
    }

    // Writes the report.
     int totalTosses = 0;
     for (Integer tossCount : tossCounts.keySet()) {
      Integer freq = tossCounts.get(tossCount);
      System.out.println(String.format("%d tosses: %d", tossCount, freq));
      totalTosses += (tossCount * freq);
    }
    System.out.println("Total tosses = " + totalTosses);
    System.out.println("Average tosses = " + totalTosses * 1.0 / people.size());
  }

  // The main function.
  // The only command line parameter specifies how many people should be simulated.
   public static void main(String[] args) {
    TillTailToss t = new TillTailToss(Integer.parseInt(args[0]));
    System.out.println("Number of rounds = " + t.doTosses());
    System.out.println("Frequencies:");
    t.report();
  }
}
// ----------------------------------------------------------------------------
```

5.2. **Results.** The results of the above program, for sample sizes 10, 100, 1000, 10000, 100000 and 1000000 are tabulated below.

n	10	100	1,000	10,000	100,000	1,000,000
$\lceil \log_2 n \rceil$	4	7	10	14	17	20
Rounds	3	8	13	15	18	22
1 toss	6	53	498	5028	50238	500633
2 tosses	0	24	259	2471	24816	249743
3 tosses	4	14	115	1317	12405	124257
4 tosses		5	71	608	6217	62797
5 tosses		1	31	301	3241	31347
6 tosses		0	12	169	1605	15648
7 tosses		1	7	78	718	7682
8 tosses		2	3	44	395	3930
9 tosses			1	16	185	2003
10 tosses			1	9	82	977
11 tosses			1	2	51	481
12 tosses			1	1	26	240
13 tosses				2	12	141
14 tosses				2	2	58
15 tosses				2	3	33
16 tosses					1	16
17 tosses					2	4
18 tosses					1	4
19 tosses						2
20 tosses						3
21 tosses						0
22 tosses						1
Total	18	191	1988	20022	199629	1999539
Average	1.8	1.91	1.988	2.0022	1.99629	1.999539

6. Further notes

This puzzle is more popularly known as *The Sultan's girl puzzle.*

In the original puzzle, a Sultan decides to increase the female population in his country, and passes a law that every woman should stop getting pregnant after she has a boy. The puzzle is to find whether this law is effective to increase the female to male ratio. (The answer is no.) This is a slight modification of the puzzle.