

538 Riddler: 29 April 2022

Colin Parker

April 30, 2022

1 Express

We can say a decimal expansion terminates if $\frac{1}{n} = \frac{p}{10^q}$ for some $p, q \in \mathbb{Z}$. This implies $np = 10^q = 2^q 5^q$, and therefore that n is divisible only by 2 and 5. Conversely, if n is divisible only by 2 and 5, we can write $n = 2^a 5^b$, $n \cdot 2^b 5^a = 10^{a+b} \implies \frac{1}{n} = \frac{2^b 5^a}{10^{a+b}}$ is a terminating decimal expansion. Therefore the only values of n for which the expansion of $1/n$ terminates are $n = 2^a 5^b$. Taking inspiration from Euler's product formula for the Riemann zeta function (with $s = 1$), we can write

$$\begin{aligned} \sum_{a,b \geq 0} \frac{1}{2^a 5^b} &= \prod_{k \in \{2,5\}} \left(1 + \frac{1}{k} + \frac{1}{k^2} + \dots\right) \\ &= \prod_{k \in \{2,5\}} \frac{1}{1 - k^{-1}} \\ &= 2 \cdot \frac{5}{4} = 2.5 \end{aligned}$$

2 Classic

This problem was pretty impenetrable in the time I spent working on it. I could see that $x \approx 1/4$, just slightly less, and that $P(\text{Nicks win}) \approx 1/8$, $P(\text{Naughts win}) \approx 3/8$, but I had a hard time proving any of these things by induction, recursion, or raw calculation. Here is the code I used to obtain these numbers:

```
import numpy as np
from collections import defaultdict

def play_game(x):
    def play_pos(t0_score, t1_score, pos):
        score_prob = 0.5
        if pos: # i.e. team 1 (Naughts) has possession
            if t0_score > t1_score:
                score_prob += x
```

```

        elif t0_score < t1_score:
            score_prob -= x
            t1_score += (np.random.random() < score_prob)
        else: # team 0 (Nicks) has possession
            if t0_score > t1_score:
                score_prob -= x
            elif t0_score < t1_score:
                score_prob += x
            t0_score += (np.random.random() < score_prob)
        return (t0_score, t1_score, 1 - pos)

game_state = play_pos(0, 0, 0)
for _ in range(1, 50):
    game_state = play_pos(*game_state)

return game_state[:-1]

results = dict()

for x in np.linspace(0.05, 0.45, 9):
    results[x] = defaultdict(int)
    for _ in range(10**4):
        result = play_game(x)
        if result[0] > result[1]: results[x][0] += 1
        elif result[0] < result[1]: results[x][1] += 1
        else: results[x]['Tie'] += 1

print(results)

```