Birthday Paradox (and Meet in the Middle!)

Ryan Chang
chang.ryan10145@gmail.com

May 9, 2022

Overview

Guiding Problem

Problem

You start at the number 3 and your goal is to reach the number 10423. You are allowed up to 100 moves where during each move you are allowed to either:

1. Multiply the current number by 21233 and take the remainder when divided by 1000000007
2. Add 1023443 to the current number and take the remainder when divided by 1000000007

Find an algorithm for your computer that can find this sequence of 100 moves in at most 5 seconds.

Birthday Paradox

- Imagine you have 30 people in a room
- One by one, you go up to people and ask for their birthday
- What's the probability that two people you ask will have the same birthday?
- Assumption: birthdays are uniformly distributed across all days of the year and ignore leap years
- The uniform assumption gives us a lower bound on our probability

Approximate Solution

- Use *complementary probability*
- With 30 people, you have $\binom{30}{2} = 435$ pairs of people
- The probability that two people don't have the same birthday is $1 - 1/365 = 364/365$
- The probability that all 435 pairs of people don't have the same birthday is: $(364/365)^{435} \approx 0.303$
- The probability that at least one pair has the same birthday is: $1 - 0.303 = \boxed{0.697}$
- This is only an **approximate solution**. Why?
- Doing $P(A \cup B) = P(A) \cdot P(B)$ works if and only if A and B are independent
- If Alice and Bob have the same birthday and Bob and Charles have the same birthday, then probability that Alice and Charles have the same birthday is 1
- **Not independent**

Actual Solution

- Use a different approach (still complementary probability)
- Consider the people one by one and find the probability that their birthday doesn't match any of the people we have seen so far

$$365/365 * 364/365 * 363/365 * \cdots = \prod_{i=0}^{29}(1 - i/365) \approx 0.2937$$

- Our probability is then $\boxed{0.7063}$
- Our approximation was very close!

Another Approximation

- Previous formula is harder to use
- Let's make it simpler!
- From Taylor Series: $e^x = 1 + x + x^2/2! + x^3/3! + \ldots$
- For small x, x^2 term is very small, so we have $e^x = 1 + x$

$$\prod_{i=0}^{29}(1 - i/365) \approx \prod_{i=0}^{29}(e^{-i/365})$$

- Multiplying adds the exponents and $\sum_{i=0}^{29} i = 435$

$$\prod_{i=0}^{29}(e^{-i/365}) = e^{-435/365} = 0.3037$$

- 1 - 0.3037 = $\boxed{0.6963}$

Theorem

Theorem (Birthday Paradox)

If you have n numbers each uniformly chosen from k values, then the probability that two numbers will have the same value is ≈ 0.5 when $n = 1.2\sqrt{k}$

- For the birthday problem, $k = 365$
- $n = 1.2 * \sqrt{365} = 22.9 \approx 23$

$$1 - \prod_{i=0}^{22}(1 - i/365) = 0.507$$

- For $k = 1000000007$, $n = 1.2\sqrt{1000000007} \approx 37947$

$$1 - \prod_{i=0}^{37947-1}(1 - i/1000000007) \approx 0.513$$

Theorem Proof

- Earlier, we discussed how we can approximate our answer with this:

$$1 - \prod_{i=0}^{n-1} (e^{-i/k})$$

- With exponent rules and applying sum of first n terms formula this is equivalent to:

$$1 - e^{-n(n-1)/2k} \approx 1 - e^{-n^2/2k}$$

Theorem Proof Continued

- We then set this equal to our desired probability 0.5 to obtain:

$$0.5 = 1 - e^{-n^2/2k} \Rightarrow n = \sqrt{2k \cdot -\ln(0.5)} = \sqrt{2\ln 2} \cdot \sqrt{k}$$

- $\sqrt{2\ln 2} \approx 1.177$ which I rounded to 1.2
- We can do this for any probability p with the general formula:

$$n = \sqrt{2\ln(1/(1-p))} \cdot \sqrt{k}$$

- For $p = 0.9$ we have $2.15\sqrt{k}$
- For $p = 0.99$ we have $3.03\sqrt{k}$
- For $p = 0.999999999$ we have $6.44\sqrt{k}$
- The above is practically guaranteed and we don't even have to do that many samples!

Guiding Problem

Problem

You start at the number 3 and your goal is to reach the number 10423. You are allowed up to 100 moves where during each move you are allowed to either:

1. Multiply the current number by 21233 and take the remainder when divided by 1000000007
2. Add 1023443 to the current number and take the remainder when divided by 1000000007

Find an algorithm for your computer that can find this sequence of 100 moves in at most 5 seconds.

Brute Force

- Try every possible sequence of moves and see if any land you there
- $2^{100} \approx 10^{30}$ total sequences (way too much)
- Observation 1: After a set of moves, we're going to reach a number and the rest of our sequence only depends on that number
- Observation 2: Many of our 2^{100} sequences are going to intersect
- If two sequences intersect, we don't need to consider both of them separately
- 1000000007 possible numbers, so 1000000007 possible paths we consider
- Still too much! Computers can typically perform around 100M operations per second

Meet in the Middle

- Let's go back to the idea of brute force
- Instead of trying all 2^{100} sequences from the beginning with the goal of reaching the end, let's try searching from both the beginning and end
- Let's try all sequences of length 50 from the beginning and all sequences of length 50 from the end and if any intersect, then we have a path from the beginning to the end
- $2^{50} * 2 \approx 2 * 10^{15}$
- A significant improvement, but not enough!
- Observations from before already got us better than this

Meet in the Middle + Birthday Paradox

- 2^{50} paths was too much
- Let's run n random paths from the beginning where n is some constant we choose
- We will hit $\approx n$ random numbers in $[0, 1000000007)$
- Let's run n random paths from the end
- We will hit $\approx n$ random numbers in $[0, 1000000007)$
- We are looking for any spot where those n random paths from our end have the same number as one of those n random paths from our beginning
- Sounds like Birthday Paradox!

Birthday Paradox Cont.

- If we choose $n = 1.2\sqrt{1000000007} \approx 37947$, then we have at least a $1/2$ chance to get a match
- In reality, the probability is probably higher (due to numbers not being uniformly random)
- We should choose a higher n to get a higher probability
- This will not find the **shortest** path, since we are just running random paths but that wasn't the goal of our problem

Python Demonstration

https://replit.com/@RyanC3/BirthdayParadoxMeetInTheMiddlemain.py

- Numbers are reduced a bit in this replit because the website is slow
- In the original presentation, I gave a live demonstration on my computer instead where I just bumped up the numbers and ran it locally

The End