Algorithms for Programming Contests SS20 - Week 09

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax,
A.R.Balasubramanian, Christoph Welzel

June 23, 2020

A: Commander-in-Chief - Sample Solution

- Use the identity $gcd(a, b, c) = gcd(gcd(a, b), c)$.
- Euclidean algorithm

B: Break In - Sample Solution

Problem

Find out the numbers Lea has to enter on the touchpads to open the door. Lea knows one of the numbers and knows that their product is always 1 mod 10^n.

B: Break In - Sample Solution

- Find $x \in [1, 10^n)$ such that $x \cdot y \equiv 1 \pmod{10^n}$.
- Observation: y and 10^n are coprime.
- Then x is the modular multiplicative inverse of y.
- Use Extended Euclidean Algorithm to find x.
- If resulting x is negative, output $10^n + x$, as $|x| \leq 10^n$.

C: Candies - Sample Solution

Problem

Lea wants to have enough candies so that no matter who shows up, she can divide them evenly.

Solution

- Iterate through all possible combinations of people (2^{15}) and compute the sum of candies consumed per round.
- Compute the LCM of those sums with the following equation: $\text{LCM}(a, b) = \frac{a \cdot b}{\text{GCD}(a,b)}$.
- Take care to first divide by the GCD as to not overflow the long!

D: N-athlon - Sample Solution

Problem

Find the biggest k such that the team decomposition Lea remembers works.

D: N-athlon - Sample Solution

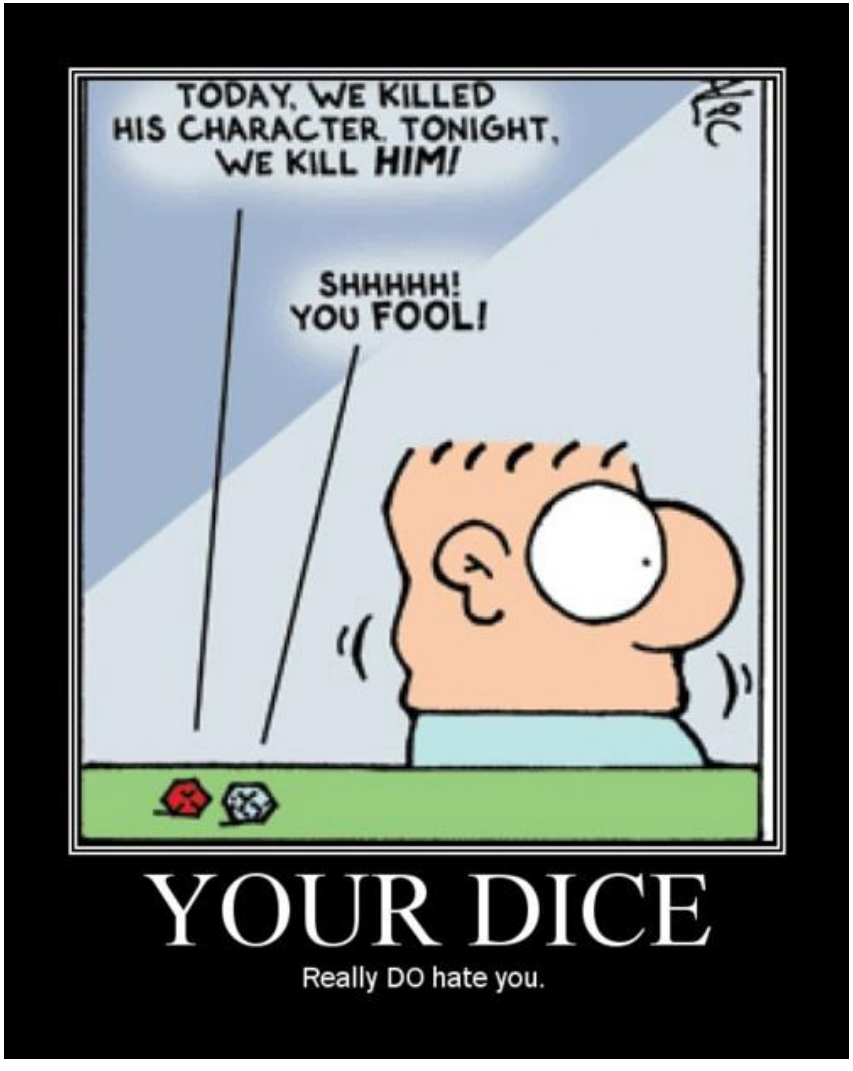
Solution

- Chinese Remainder Theorem
- Find the largest possible answer smaller than k
- Consistens: Same amount of teams, same remainder - remove the equation
- Inconsistent numbers may make the problem unsolvable
- Inconsistent: Same amount of teams, different remainder
- Caveats:
  - Large numbers: Either use mod N often enough or use BigInteger
  - Care for edge cases: 0 people invited, 0 games played, ...
  - Two primes are NOT coprime if they are the same number.

E: Vaults & Vampires - Sample Solution

- Always save numerator and denominator for rationals, use BigIntegers.
- Iterate through dice one by one and save probability for each sum.
- $prob[i][j]$: probability of sum j after i dice have been rolled.
- Set initial values for the first die.
- $prob[i][j] = \sum_{k=1}^{size[i]} prob[i-1][j-k]/size[i]$ (ignore negative indices)
- Even faster: Sum up only the numerators during the dp, divide the result by the product of the sizes in the end.

E: Vaults & Vampires - Sample Solution



F: Game Show - Sample Solution

- Write a class for rationals with the desired functionalities.
- Use BigInteger for numerator and denominator.
- Summing the terms with a loop takes too long!
- Use the formula

$$\sum_{i=1}^{n} r^i = \frac{r(1-r^n)}{1-r}.$$

G: Soft Skills - Sample Solution

Problem

Compute the amount of possible combinations of Soft Skill Course Titles Lea can accumulate.

Solution

Compute $\binom{n}{k} \mod x$.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

G: Soft Skills - Sample Solution

- Problems: big numbers, can't divide modulo x if x is not prime
- One way: compute it using factorials, split each number into prime divisors, carefully count prime numbers in the numerator and denominator

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

G: Soft Skills - Sample Solution

- Second way: compute $\binom{n}{k} \mod p$ for all prime divisors p of x, assemble solution of $\binom{n}{k} \mod x$ by chinese remaindering
- Compute $\binom{n}{k} \mod p$ by removing all factors p and counting them on their own
- Dividing by a number modulo p is possible if p is a prime, use the extended euclidean algorithm to compute $a \cdot p + b \cdot y = 1$, b is the inverse of y modulo p