Discrete Math - Bashing

Jerry Huang
Assistant Lecturer: Akshaj Kadaveru

special thanks to codeforces

Feb 12 2014

1 Introduction

A rarely applied field of number theory, yet so powerful to make even an China TST problem trivial, Number Bashing. What is number bashing, you might ask, as even a excellent mathematician as yourself might be unfamiliar with this term. In Geometry, there are many ways to solve a problem, and these various methods are divided into two major subcategories, named synthetically and bash. How can we change this such that we could bash those annoying number theory problems? The answer is Number Bashing. Note that because Pure Number Bashing heavily relies on the use of computers, this method will not be useful in contests like the AIME. We will first look at the simple, yet effective, brute force method.

2 A Few Restrictions

For the rest of the problems, assume you have a superpower that lets you perform about $7 \cdot 10^7$ computations. This means that the number of computations required to obtain the answer should be about 10^8. For example, if a question asked "Find $10^7 \pmod{17}$", an acceptable solution would be to let a variable $a = 1$ and perform the assignment $a = (10a)\%17$ many times.

The reason we make this restriction is that on many programming competitions, time is restricted to one second, in which programming languages such as Java can perform about this many computations.

3 Brute Force

(2003 AMC 10A Problem 25) Let n be a 5-digit number, and let q and r be the quotient and remainder, respectively, when n is divided by 100. For how many values of n is $q+r$ divisible by 11?

Darn, this problem requires the divisibility rule of 11, but lol oops I forgot it! Now what can I do? Well, I know that to be a answer, $9999 < n < 100000$, and that $q+r \equiv 0 \pmod{11}$, where $q = \lfloor \frac{n}{100} \rfloor$ and $n \equiv r \pmod{100}$. But the conditions are extremely small (remember, we can perform about 10^8 computations!!) Thus a simple brute-force check will work. The python code is shown here:

```
ans = 0
for i in range(10000,100000):
    if ((i / 100) + (i % 100)) % 11 ==  0:
        ans = ans + 1
print ans
```

giving an answer of 8181. ■

Okay, that was a very uninstructive problem, but it illustrates that when possible, brute force is the best way to do a problem.

4 A Few Problems

(Remember your superpower; you don't actually have to solve the problem, just devise an algorithm that works)

1. (AlcumusGuy's Mock AMC 10 #24) Denote by $\lfloor x \rfloor$ the greatest integer that is less than x. For example, $\lfloor 2.5 \rfloor = 2$ and $\lfloor -\pi \rfloor = -4$. How many distinct values are in the set $\left\{ \left\lfloor \frac{1^2}{2015} \right\rfloor, \left\lfloor \frac{2^2}{2015} \right\rfloor, \left\lfloor \frac{3^2}{2015} \right\rfloor, \dots \left\lfloor \frac{2015^2}{2015} \right\rfloor \right\}$?

2. (1990 AIME Problem 13) Let $T = \{9^k : k \text{ is an integer}, 0 \leq k \leq 4000\}$.

Given that 9^{4000} has 3817 digits and that its first (leftmost) digit is 9, how many elements of T have 9 as their leftmost digit?

3. (China TST) For a positive integer M, if there exist integers a, b, c and d so that:

$$M \leq a \leq b \leq c \leq d \leq M + 49, \qquad ad = bc$$

then we call M a GOOD number, if not then M is BAD. Please find the greatest GOOD number and the smallest BAD number.

4. (Project Euler) The number of divisors of 120 is 16. In fact 120 is the smallest number having 16 divisors. Find the smallest number with 2^{500500} divisors. Give your answer modulo 500500507.

5 Dynamic Number Bashing

Basically Dynamic Number Bashing is just dynamic programming (dp) applied to the context of Number Bashing. If you do not know what dp is, it is essentially the trade off of memory for speed.

6 The first dimension

You are given a tetrahedron. Let's mark its vertices with letters A, B, C and D correspondingly. An ant is standing in the vertex D of the tetrahedron. The ant is quite active and he wouldn't stay idle. At each moment of time he makes a step from one vertex to another one along some edge of the tetrahedron. The ant just can't stand on one place. You do not have to do much to solve the problem: your task is to count the number of ways in which the ant can go from the initial vertex D to itself in exactly 10^7 steps. In other words, you are asked to find out the number of different cyclic paths with the length of 10^7 from vertex D to itself.

Please work on this problem for 5 minutes, and then we will ask people to present their ideas.

7 Another example (2D)

Bob wants to put a new bargaining table in his office. To do so he measured the office room thoroughly and drew its plan: Bob's office room is a rectangular room 100 by 100 meters Each square meter of the room is either occupied by some furniture, or free. A bargaining table is rectangular, and should be placed so, that its sides are parallel to the office walls. Bob doesn't want to change or rearrange anything, that's why all the squares that will be occupied by the table should be initially free. Bob wants the new table to sit as many people as possible, thus its perimeter should be maximal. Devise an algorithm for Bob to find out the maximum possible perimeter of a bargaining table for his office, for any starting arrangement of the furniture.

Solution: In this problem one should find the maximal perimeter of a rectangle that contains no '1'. Define these rectangles "correct". To solve a problem you are to check each possible rectangle for correctness and calculate its perimeter. The easiest way to check all rectangles is using 6 nested cycles. Using 4 of them you fix the coordinates while other 2 will look for '1'. So the complexity is $O((nm)^3)$. This is too slow for us, as $(100 \cdot 100)^3 = 10^{12}$ which is way too big.

One may interest a faster solution. Using simple DP solution one can get a solution with an $O((nm)^2)$ complexity. It's clear, that rectangle with coordinates (x_1, y_1, x_2, y_2) is correct if and only if rectangles $(x_1, y_1, x_2 - 1, y_2)$ and $(x_1, y_1, x_2, y_2 - 1)$ are correct, and board$[x_2][y_2] =' 0'$. So each of rectangles can be checked in $O(1)$ and totally there will be $O((nm)^2)$ operations.

8 Codeforces Problem (27E)

Given the number $n \geq 1$, find the smallest positive integer which has exactly n divisors. It is guaranteed that for the given n the answer will not exceed 2^{18}.

Consider the number, that is our answer and factorize it. We will get such product $p_1^{a1} \cdot p_2^{a2} \cdot \ldots \cdot p_k^{ak}$. Product through each i $a_i + 1$ will be the number of divisors. So, if we will take first 18 prime numbers, their product will have 2^{18} divisors. This means that we need only first 18 primes to build our answer if $n \leq 2^{18}$.

Let's do it with dynamic programming: $d[i][j]$ - the minimal number with i divisors that can be built with first j prime numbers. To calculate the answer for state (i, j) let's look over all powers of jth prime number in the answer. If jth prime number has power k in the answer, than $d[i][j] = d[i/(k+1)][j-1] \cdot \text{prime}[j]^k$. For each power of jth prime we must select the power, that gives us minimal $d[i][j]$.

9 Dynamic Number Bashing Problems

Please note that these problems might be able to be solved with synthetical number theory, but attempting them in such a fashion might lead to nowhere.

1. A table composed of $1000x1000$ cells, each having a certain quantity of apples, is given. You start from the upper-left corner. At each step you can go down or right one cell. Find the maximum number of apples you can collect.

2. You are given two arbitrary strings of letters, of length 10^3 each. Devise an algorithm to find the length of the longest common sub-sequence of both of these strings. For example, 1123 is a subsequence of both 15120034 and 110230 because the digits $1, 1, 2, 3$ appear in order in both strings.

3. (Codeforces) There is a square matrix of size 1000 by 1000, consisting of non-negative integer numbers. You should find a path through it such that the path starts in the upper left cell of the matrix; each following cell is to the right or down from the current cell; and the path ends in the bottom right cell. Moreover, if we multiply together all the numbers along the way, the result should be the least "round". In other words, it should end in the least possible number of zeros.

10 Extra Practice/Problems

1. Codeforces, Project Euler
2. Problems in Elementary Number Theory (PEN)