A SOTA Coding Problem
View on GitHub

Rishit Dagli
Twitter | GitHub | Medium | LinkedIn

June 4, 2020

Introduction

I recently came across a State of the Art (SOTA) problem according to me. How I define a problem to be really good-

- While reading the problem it seems to be very easy
- It becomes hard while you try and code it (By logic and not by knowledge of coding)
- The optimal solution is just a few lines of code

I do not claim that the problem has been made by me in any manner. I found this question while attempting Google FooBar.

Dodge the Lasers! (The Problem)

Someone who you earlier worked for has sent her elite fighter pilot squadron after you - and they've opened fire!
Fortunately, you know something important about the ships trying to shoot you down. Back when you were still her assistant, she asked you to help program the aiming mechanisms for the ships. They undergo rigorous testing procedures, but you were still able to slip in a subtle bug. The software works as a time step simulation: if it is tracking a target that is accelerating away at 45 **degrees, the software will consider the targets acceleration to be equal to the** $\sqrt{2}$**, adding the calculated result to the targets end velocity at each timestep. However, thanks to your bug, instead of storing the result with proper precision, it will be truncated to an integer before adding the new velocity to your current position. This means that instead of having your correct position, the targeting software will erringly report your position as-**

$$\sum_{i=1}^{n} \lfloor i \cdot \sqrt{2} \rfloor$$

not far enough off to fail her testing, but enough that it might just save your life.

If you can quickly calculate the target of the starfighters' laser beams to know how far off they'll be, you can trick them into shooting an asteroid, releasing dust, and concealing the rest of your escape. Write a function $solution(str_n)$ which, given the string representation of an integer n, returns the sum of

$$\lfloor 1 \cdot \sqrt{2} \rfloor + \lfloor 2 \cdot \sqrt{2} \rfloor + \ldots + \lfloor n \cdot \sqrt{2} \rfloor$$

as a string. That is, for every number i in the range 1 to n, it adds up all of the integer portions of $i \cdot \sqrt{2}$.
For example, if str_n was "5", the solution would be calculated as-
$floor(1 * sqrt(2)) +$
$floor(2 * sqrt(2)) +$
$floor(3 * sqrt(2)) +$
$floor(4 * sqrt(2)) +$
$floor(5 * sqrt(2))$
$= 1 + 2 + 4 + 5 + 7 = 19$
so the function would return "19"

Twist

n can have values upto 10^100 which means it can have numbers with 101 digits! Try solving this problem yourself first before heading to the solution.
PS: If it was as easy as a for loop I would not have written this in the first place

Hints

This seems like a very easy problem. If you remove the story line, all you need to calculate is-

$$\sum_{i=1}^{n} \lfloor i \cdot \sqrt{2} \rfloor$$

The only twist is you cant use recursion for each element or use a for loop as the numbers are too big. You need to try and simplify the expression.

Solution

Just some convention-
This type of sequence is called a Beatty sequence. Beatty sequence is the sequence of integers found by taking the floor of the positive multiples of a positive irrational number. However, you do not require any knowledge of it to solve the problem.

Let's define a function $\beta_r^{(i)} = \lfloor i \cdot r \rfloor$ for some irrational number and

$$S(r, n) = \sum_{i=1}^{n} \beta_r^{(i)}$$

if $r \geq 2$ we say $p = r - 1$ and we can also say-

$$\begin{aligned} S(p,n) &= S(s,n) + \sum_{i=1}^{n} \cdot i \\ &= S(s,n) + \frac{n(n+1)}{2} \end{aligned}$$

Also, if $1 < r < 2$ and $\frac{1}{p} + \frac{1}{r} = 1$ then sequences β_r and β_p for $n \geq 1$ partition $\mathbb{N}$
So,

$$S(r,n) + S(s, \lfloor \frac{\mathcal{B}_r^{(n)}}{s} \rfloor) = \sum_{i}^{\mathcal{B}_r^{(n)}} i = \frac{\mathcal{B}_r^{(n)}(\mathcal{B}_r^{(n)} + 1)}{2}$$

And also

$$\begin{aligned} \lfloor \frac{\mathcal{B}_r^{(n)}}{s} \rfloor &= \lfloor \mathcal{B}_r^{(n)}(1 - \frac{1}{r}) \rfloor \\ &= \mathcal{B}_r^{(n)} - \lceil \frac{\mathcal{B}_r^{(n)}}{r} \rceil \\ &= \mathcal{B}_r^{(n)} - n \end{aligned}$$

Then, letting $n' = \lfloor (r-1)n \rfloor = \mathcal{B}_{r-1}^{n}$ we have

$$S(r,n) = \frac{\mathcal{B}_r^{(n)}(\mathcal{B}_r^{(n)} + 1)}{2} - S(p,n') = \frac{(n+n')(n+n'+1)}{2} - S(p,n')$$

Back to the problem, we have $p = \sqrt{2}$, so we start with $p = 2 + \sqrt{2}$. We can get a recurrence formula. Let $n' = \lfloor (\sqrt{2} - 1)n \rfloor$

$$\begin{aligned} S(\sqrt{2},n) &= \frac{(n+n')(n+n'+1)}{2} - S(2+\sqrt{2},n') \\ &= \frac{(n+n')(n+n'+1)}{2} - (n'(n'+1) - S(\sqrt{2},n')) \\ &= nn' + \frac{n(n+1)}{2} - \frac{n'(n'+1)}{2} - S(\sqrt{2},n') \end{aligned}$$

And we solved the problem, got an optimal way to solve it :)

In Code

Let us now take a look at implementing this in code. We will just convert all what we just talked about to code.

```python
minus_sqrt2 = 4142135623730950488016887242096980785696718753769480731766797379907324784621070388503875343276415727
def n1(n):
    return (minus_sqrt2 * n) // (10 ** 100)
def s(n):
    if n == 1:
        return 1
    if n < 1:
        return 0
    return n * n1(n) + n * (n + 1) / 2 - n1(n) * (n1(n) + 1) / 2 - s(n1(n))
def solution(n):
    return str(s(int(n)))
```

In case you aren't able to view the rendered code block, navigate down to "Try out yourself". You can see it is just 11 lines of code to tackle this problem! I think this is pretty cool, as just your logic was what counted here and not knowledge of a programming language.

Try out for yourself

You can try out this code for yourself using these links-

- Colab environment - Requires a Google Account (Recommended)
- Repl.it environment

About Me

Hi everyone I am Rishit Dagli. This was my first time writing a blog on a specific problem which I feel could help others develop their logic. If you like this one and would love to see more of these kind from me, consider starring the repo
If you want to ask me some questions, report any mistake, suggest improvements, give feedback you are free to do so by emailing me at-

- rishit.dagli@gmail.com
- hello@rishit.tech