441. Arranging Coins

March 25, 2020 | 70.2K views



(1) (1) (in)

You have a total of n coins that you want to form in a staircase shape, where every k-th row must have exactly k coins.

Given n, find the total number of **full** staircase rows that can be formed.

n is a non-negative integer and fits within the range of a 32-bit signed integer.

Example 1:

Example 2:

Solution

Approach 1: Binary Search

This question is easy in a sense that one could run an **exhaustive iteration** to obtain the result. That could work, except that it would run out of time when the input becomes too large. So let us take a step back to look at the problem, before rushing to the implementation.

Assume that the answer is k, i.e. we've managed to complete k rows of coins. These completed rows contain in total $1+2+\ldots+k=\frac{k(k+1)}{2}$ coins.

We could now reformulate the problem as follows:

```
Find the maximum k such that rac{k(k+1)}{2} \leq N.
```

The problem seems to be one of those **search** problems. And instead of naive iteration, one could resort to another more efficient algorithm called **binary search**, as we can find in another similar problem called **search** insert position.

Implementation

```
Сору
Java Python
1 class Solution:
      def arrangeCoins(self, n: int) -> int:
         left, right = 0, n
          while left <= right:
             k = (right + left) // 2
             curr = k * (k + 1) // 2
             if curr == n:
                return k
9
             if n < curr:
10
                right = k - 1
              else:
11
                 left = k + 1
13
          return right
```

Complexity Analysis

- Time complexity : $\mathcal{O}(\log N)$.
- Space complexity : $\mathcal{O}(1)$.

Approach 2: Math

If we look deeper into the formula of the problem, we could actually solve it with the help of mathematics, without using any iteration.

 $k(k+1) \leq 2N$

As a reminder, the constraint of the problem can be expressed as follows:

 $\left(k+\frac{1}{2}\right)^2-\frac{1}{4}\leq 2N$

$$k=\left[\sqrt{2N+rac{1}{4}}-rac{1}{2}
ight]$$

that results in the following answer:

```
Java Python

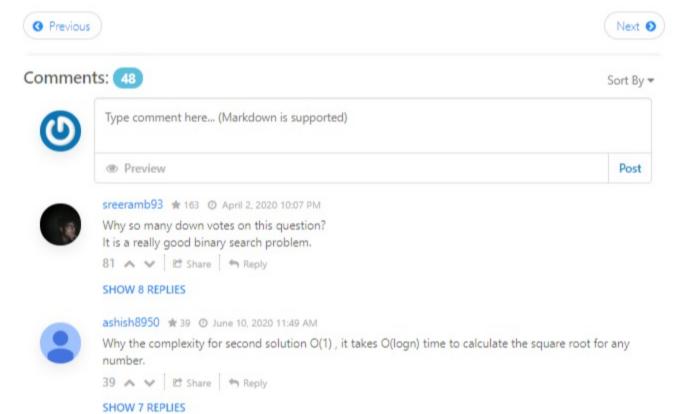
1 class Solution:
2 def arrangeCoins(self, n: int) -> int:
3 return (int)((2 * n + 0.25)**0.5 - 0.5)
```

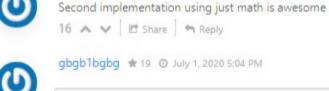
Complexity Analysis \bullet Time complexity : $\mathcal{O}(1)$.

Implementation

- Space complexity : $\mathcal{O}(1)$.
- space complexity . O(1)

Rate this article: * * * * *





Subho_Kundu ★ 26 ② July 1, 2020 12:47 PM

theNoobCoder # 4 @ July 2, 2020 12:00 AM



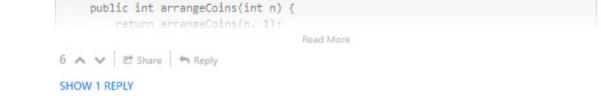
First problem on leetcode. Took 4 mins for first submission. But Time out. Then Took 4 hours to figure out only that I was thinking too high. ;D. Finally accepted.

4 A V B Share Reply

SHOW 2 REPLIES

rkothakapu ★ 6 ② July 1, 2020 6:26 PM Java - Using Recursion

class Solution {



Amazed to see the way binary search is used to solve this problem.

SHOW 1 REPLY

bhruti1234bhruti * 8 ② July 1, 2020 6:11 PM

Always thought that binary search will only be used if there is "sorted " word involved in the question.

6 ∧ ∨ ₾ Share ♠ Reply SHOW 1 REPLY

very good question.



Ardh1999 ★ 13 ② July 1, 2020 1:54 PM (sqrt(1+(8*n))-1)/2

SHOW 5 REPLIES



Interesting alternative solutions.

