441. Arranging Coins

March 25, 2020 | 70.2K views

Average Rating: 4.69 (171 votes)

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:

```
n = 5

The coins can form the following rows:
¤
¤ ¤
¤ ¤

Because the 3rd row is incomplete, we return 2.
```

Example 2:

```
n = 8

The coins can form the following rows:
¤
¤ ¤
¤ ¤ ¤
¤ ¤

Because the 4th row is incomplete, we return 3.
```

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 $1 + 2 + ... + k = \frac{k(k+1)}{2}$ coins.

We could now reformulate the problem as follows:

> Find the maximum k such that $\frac{k(k+1)}{2} \leq N$.

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

Java | Python — Copy

```python
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
            if n < curr:
                right = k - 1
            else:
                left = k + 1
        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.

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

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

This could be solved by completing the square technique,

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

that results in the following answer:

$$k = \left\lfloor \sqrt{2N + \frac{1}{4}} - \frac{1}{2} \right\rfloor$$

Implementation

Java | Python — Copy

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

Complexity Analysis

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

Rate this article:

Previous Next

Comments: 48 — Sort By

Type comment here... (Markdown is supported)

Preview | Post

sreeramb93 ★ 163 April 2, 2020 10:07 PM
Why so many down votes on this question?
It is a really good binary search problem.
81 Share | Reply

SHOW 8 REPLIES

ashish8950 ★ 39 June 10, 2020 11:49 AM
Why the complexity for second solution O(1) , it takes O(logn) time to calculate the square root for any number.
39 Share | Reply

SHOW 7 REPLIES

Subho_Kundu ★ 26 July 1, 2020 12:47 PM
Second implementation using just math is awesome
16 Share | Reply

gbgb1bgbg ★ 19 July 1, 2020 5:04 PM

```
public:
    int arrangeCoins(int n) {
        int count = 0;
        int [ ...
```

Read More

19 Share | Reply

SHOW 1 REPLY

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 Share | Reply

SHOW 2 REPLIES

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

```
class Solution {
    public int arrangeCoins(int n) {
        return arrangeCoins(n, 1);
```

Read More

6 Share | Reply

SHOW 1 REPLY

bhruti1234bhruti ★ 8 July 1, 2020 6:11 PM
Always thought that binary search will only be used if there is "sorted " word involved in the question. Amazed to see the way binary search is used to solve this problem.
very good question.
6 Share | Reply

SHOW 1 REPLY

filipre ★ 14 July 1, 2020 1:32 PM
How does the second solution make sure, that we don't run into numeric issues? What happens, if for some weird edge case the `sqrt()` function returns something like `x.4999999999` ?
3 Share | Reply

SHOW 5 REPLIES

Ardh1999 ★ 13 July 1, 2020 1:54 PM
(sqrt(1+(8*n))-1)/2
as simple is the sol^n
4 Share | Reply

SHOW 3 REPLIES

tivan ★ 8 July 1, 2020 8:28 PM
Interesting alternative solutions.
2 Share | Reply

1 2 3 4 5