



链滴

# [每日 LeetCode] 441. Arranging Coins

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1561564275752>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 441. Arranging Coins](#)

### Description:

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:

```

x
x x
x x
```

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

### Example 2:

$n = 8$

The coins can form the following rows:

```

x
x x
x x x
x x
```

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

---

思路：本题要求排列硬币，求最大阶梯数。

- 思路一：简单粗暴的方法啊，从第一行开始，一行一行的从 $n$ 中减去，如果此时剩余的硬币没法满下一行需要的硬币数了，返回当前行数即可。
- 思路二：看到其他的解法，直接使用求根公式求解。列出公式 $n = (1 + x) * x / 2$ ，用一元二次方程求根公式可以得到 $x = (-1 + \sqrt{8 * n + 1}) / 2$ ，然后取整后就是能填满的行数。

---

C++代码（思路一）

```
class Solution {
public:
    int arrangeCoins(int n) {
        int cur = 1, rem = n - 1;
        while (rem >= cur + 1) {
            ++cur;
            rem -= cur;
        }
    }
};
```

```
    }  
    return n == 0 ? 0 : cur;  
  }  
};
```

运行时间: 12ms

运行内存: 8.3M

---

C++代码 (思路二)

```
class Solution {  
public:  
    int arrangeCoins(int n) {  
        return (int)((-1 + sqrt(1 + 8 * (long)n)) / 2);  
    }  
};
```

运行时间: 4ms

运行内存: 8.3M

---