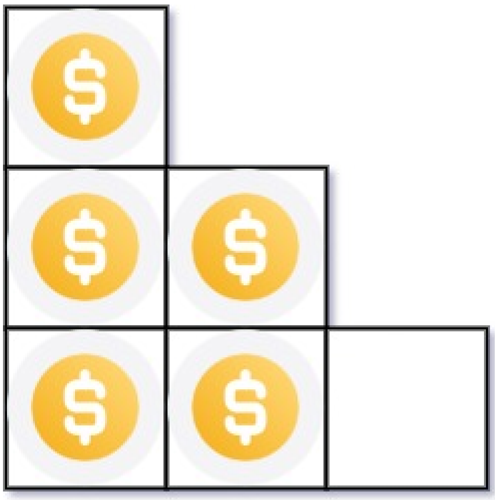
441. Arranging Coins

You have `n` coins and you want to build a staircase with these coins. The staircase consists of `k` rows where the `ith` row has exactly `i` coins. The last row of the staircase **may be** incomplete.

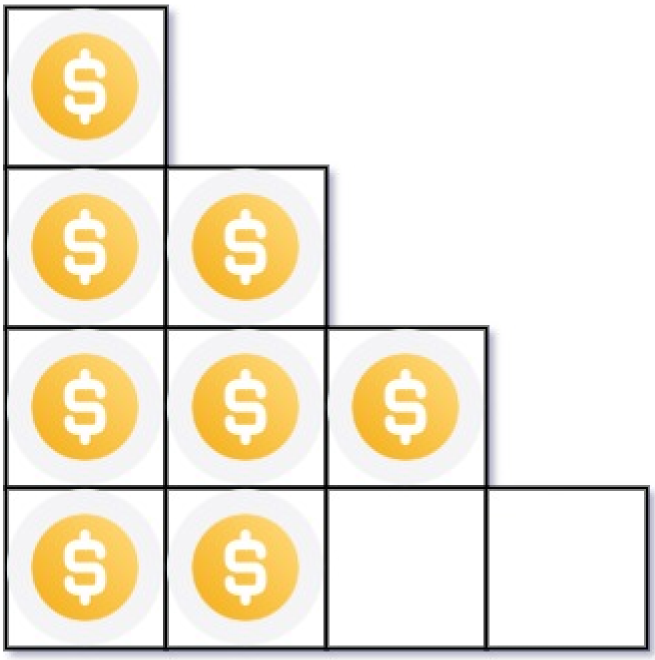
Given the integer `n`, return *the number of **complete rows** of the staircase you will build*.

Example 1:



Input: n = 5
Output: 2
Explanation: Because the 3rd row is incomplete, we return 2.

Example 2:



Input: n = 8

Output: 3

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

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

        #Second approach is very good BS
#         lo = 1
#         hi = n

#         while lo<=hi:
#             mid = (lo+hi)//2
#             temp = mid*(mid+1)//2
#             if temp==n:
#                 return mid
#             elif temp<n:
#                 lo = mid+1
#             else:
#                 hi = mid-1
#         return lo-1
```