

Problem 2139: Minimum Moves to Reach Target Score

Problem Information

Difficulty: **Medium**

Acceptance Rate: 51.92%

Paid Only: No

Tags: Math, Greedy

Problem Description

You are playing a game with integers. You start with the integer `1` and you want to reach the integer `target`.

In one move, you can either:

Increment the current integer by one (i.e., $x = x + 1$). **Double** the current integer (i.e., $x = 2 * x$).

You can use the **increment** operation **any** number of times, however, you can only use the **double** operation **at most** `maxDoubles` times.

Given the two integers `target` and `maxDoubles`, return the minimum number of moves needed to reach `target` starting with `1`.

Example 1:

Input: `target = 5, maxDoubles = 0` **Output:** `4` **Explanation:** Keep incrementing by 1 until you reach target.

Example 2:

Input: `target = 19, maxDoubles = 2` **Output:** `7` **Explanation:** Initially, $x = 1$ Increment 3 times so $x = 4$ Double once so $x = 8$ Increment once so $x = 9$ Double again so $x = 18$ Increment once so $x = 19$

****Example 3:****

****Input:**** target = 10, maxDoubles = 4 ****Output:**** 4 ****Explanation:********* Initially, x = 1
Increment once so x = 2 Double once so x = 4 Increment once so x = 5 Double again so x = 10

****Constraints:****

*`1 <= target <= 109` *`0 <= maxDoubles <= 100`

Code Snippets

C++:

```
class Solution {  
public:  
    int minMoves(int target, int maxDoubles) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int minMoves(int target, int maxDoubles) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def minMoves(self, target: int, maxDoubles: int) -> int:
```