

# Problem 1936: Add Minimum Number of Rungs

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 43.64%

**Paid Only:** No

**Tags:** Array, Greedy

## Problem Description

You are given a **strictly increasing** integer array `rungs` that represents the **height** of rungs on a ladder. You are currently on the **floor** at height `0`, and you want to reach the last rung.

You are also given an integer `dist`. You can only climb to the next highest rung if the distance between where you are currently at (the floor or on a rung) and the next rung is **at most** `dist`. You are able to insert rungs at any positive **integer** height if a rung is not already there.

Return **\_the minimum number of rungs that must be added to the ladder in order for you to climb to the last rung.\_**

**Example 1:**

**Input:** rungs = [1,3,5,10], dist = 2 **Output:** 2 **Explanation:** You currently cannot reach the last rung. Add rungs at heights 7 and 8 to climb this ladder. The ladder will now have rungs at [1,3,5,7,8,10].

**Example 2:**

**Input:** rungs = [3,6,8,10], dist = 3 **Output:** 0 **Explanation:** This ladder can be climbed without adding additional rungs.

**Example 3:**

**\*\*Input:\*\*** rungs = [3,4,6,7], dist = 2 **\*\*Output:\*\*** 1 **\*\*Explanation:\*\*** You currently cannot reach the first rung from the ground. Add a rung at height 1 to climb this ladder. The ladder will now have rungs at [1,3,4,6,7].

**\*\*Constraints:\*\***

\* `1 <= rungs.length <= 105` \* `1 <= rungs[i] <= 109` \* `1 <= dist <= 109` \* `rungs` is strictly increasing\*.

## Code Snippets

### C++:

```
class Solution {  
public:  
    int addRungs(vector<int>& rungs, int dist) {  
  
    }  
};
```

### Java:

```
class Solution {  
public int addRungs(int[] rungs, int dist) {  
  
}  
}
```

### Python3:

```
class Solution:  
    def addRungs(self, rungs: List[int], dist: int) -> int:
```