

## 5814. Add Minimum Number of Rungs

[My Submissions \(/contest/weekly-contest-250/problems/add-minimum-number-of-rungs/submissions/\)](/contest/weekly-contest-250/problems/add-minimum-number-of-rungs/submissions/)[Back to Contest \(/contest/weekly-contest-250/\)](/contest/weekly-contest-250/)

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.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

### 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]`.

### Example 4:

**Input:** `rungs = [5]`, `dist = 10`

**Output:** 0

**Explanation:**

This ladder can be climbed without adding additional rungs.

### Constraints:

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

JavaScript




```
1 ▼ const addRungs = (a, dist) => {
2     let n = a.length;
3     let res = 0;
4     if (a[0] > dist) res += a[0] % dist == 0 ? a[0] / dist - 1 : a[0] / dist >> 0;
5     // pr(res);
6 ▼   for (let i = 1; i < n; i++) {
7       let diff = a[i] - a[i - 1];
8       if (diff <= dist) continue;
9       // pr(diff, dist)
10      let t = diff % dist == 0 ? diff / dist - 1 : diff / dist >> 0;
11      // pr("t", t);
12      res += t;
13  }
14  return res;
15  };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (</submissions/detail/524201398/>) ?[More Details >](/submissions/detail/524201398/)

Share your acceptance!

&lt; 1

Copyright © 2021 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs\)](/jobs/) | [Bug Bounty \(/bugbounty\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student\)](/student/) | [Terms \(/terms\)](/terms/) |[Privacy Policy \(/privacy\)](/privacy/) [United States \(/region\)](/region/)