Problem 1340: Jump Game V

Problem Information

Difficulty: Hard
Acceptance Rate: 64.52%
Paid Only: No
Tags: Array, Dynamic Programming, Sorting

Problem Description

Given an array of integers `arr` and an integer `d`. In one step you can jump from index `i` to index:

* `i + x` where: `i + x < arr.length` and ` 0 < x <= d`. * `i - x` where: `i - x >= 0` and ` 0 < x <= d`.

In addition, you can only jump from index `i` to index `j` if `arr[i] > arr[j]` and `arr[i] > arr[k]` for all indices `k` between `i` and `j` (More formally `min(i, j) < k < max(i, j)`).

You can choose any index of the array and start jumping. Return _the maximum number of indices_ you can visit.

Notice that you can not jump outside of the array at any time.

Example 1:

![](https://assets.leetcode.com/uploads/2020/01/23/meta-chart.jpeg)

Input: arr = [6,4,14,6,8,13,9,7,10,6,12], d = 2 **Output:** 4 **Explanation:** You can start at index 10. You can jump 10 --> 8 --> 6 --> 7 as shown. Note that if you start at index 6 you can only jump to index 7. You cannot jump to index 5 because 13 > 9. You cannot jump to index 4 because index 5 is between index 4 and 6 and 13 > 9. Similarly You cannot jump from index 3 to index 2 or index 1.

Example 2:

Input: arr = [3,3,3,3,3], d = 3 **Output:** 1 **Explanation:** You can start at any index. You always cannot jump to any index.

Example 3:

Input: arr = [7,6,5,4,3,2,1], d = 1 **Output:** 7 **Explanation:** Start at index 0. You can visit all the indicies.

Constraints:

* `1 <= arr.length <= 1000` * `1 <= arr[i] <= 105` * `1 <= d <= arr.length`

Code Snippets

C++:

```
class Solution {
public:
int maxJumps(vector<int>& arr, int d) {

}
};
```

Java:

```
class Solution {
public int maxJumps(int[] arr, int d) {

}
}
```

Python3:

```
class Solution:
def maxJumps(self, arr: List[int], d: int) -> int:
```