Problem 1718: Construct the Lexicographically Largest Valid Sequence

Problem Information

Difficulty: Medium
Acceptance Rate: 72.88%
Paid Only: No
Tags: Array, Backtracking

Problem Description

Given an integer `n`, find a sequence with elements in the range `[1, n]` that satisfies all of the following:

* The integer `1` occurs once in the sequence. * Each integer between `2` and `n` occurs twice in the sequence. * For every integer `i` between `2` and `n`, the **distance** between the two occurrences of `i` is exactly `i`.

The **distance** between two numbers on the sequence, `a[i]` and `a[j]`, is the absolute difference of their indices, `|j - i|`.

Return _the**lexicographically largest** sequence_ _. It is guaranteed that under the given constraints, there is always a solution._

A sequence `a` is lexicographically larger than a sequence `b` (of the same length) if in the first position where `a` and `b` differ, sequence `a` has a number greater than the corresponding number in `b`. For example, `[0,1,9,0]` is lexicographically larger than `[0,1,5,6]` because the first position they differ is at the third number, and `9` is greater than `5`.

Example 1:

Input: n = 3 **Output:** [3,1,2,3,2] **Explanation:** [2,3,2,1,3] is also a valid sequence, but [3,1,2,3,2] is the lexicographically largest valid sequence.

Example 2:

Input: n = 5 **Output:** [5,3,1,4,3,5,2,4,2]

Constraints:

* `1 <= n <= 20`

Code Snippets

C++:

```
class Solution {
public:
vector<int> constructDistancedSequence(int n) {

}
};
```

Java:

```
class Solution {
public int[] constructDistancedSequence(int n) {

}
}
```

Python3:

```
class Solution:
def constructDistancedSequence(self, n: int) -> List[int]:
```