Problem 3470: Permutations IV

Problem Information

Difficulty: Hard
Acceptance Rate: 33.16%
Paid Only: No
Tags: Array, Math, Combinatorics, Enumeration

Problem Description

Given two integers, `n` and `k`, an **alternating permutation** is a permutation of the first `n` positive integers such that no **two** adjacent elements are both odd or both even.

Return the **k-th** **alternating permutation** sorted in _lexicographical order_. If there are fewer than `k` valid **alternating permutations** , return an empty list.

Example 1:

Input: n = 4, k = 6

Output: [3,4,1,2]

Explanation:

The lexicographically-sorted alternating permutations of `[1, 2, 3, 4]` are:

1. `[1, 2, 3, 4]` 2. `[1, 4, 3, 2]` 3. `[2, 1, 4, 3]` 4. `[2, 3, 4, 1]` 5. `[3, 2, 1, 4]` 6. `[3, 4, 1, 2]` <- 6th permutation 7. `[4, 1, 2, 3]` 8. `[4, 3, 2, 1]`

Since `k = 6`, we return `[3, 4, 1, 2]`.

Example 2:

Input: n = 3, k = 2

Output: [3,2,1]

Explanation:

The lexicographically-sorted alternating permutations of `[1, 2, 3]` are:

1. `[1, 2, 3]` 2. `[3, 2, 1]` <- 2nd permutation

Since `k = 2`, we return `[3, 2, 1]`.

Example 3:

Input: n = 2, k = 3

Output: []

Explanation:

The lexicographically-sorted alternating permutations of `[1, 2]` are:

1. `[1, 2]` 2. `[2, 1]`

There are only 2 alternating permutations, but `k = 3`, which is out of range. Thus, we return an empty list `[]`.

Constraints:

* `1 <= n <= 100` * `1 <= k <= 1015`

Code Snippets

C++:

```
class Solution {
public:
vector<int> permute(int n, long long k) {

}
};
```

Java:

```
class Solution {
public int[] permute(int n, long k) {

}
}
```

Python3:

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