1415. The k-th Lexicographical String of All Happy Strings of Length n

A **happy string** is a string that:

- consists only of letters of the set `['a', 'b', 'c']`.
- `s[i] != s[i + 1]` for all values of `i` from `1` to `s.length - 1` (string is 1-indexed).

For example, strings **"abc", "ac", "b"** and **"abcbabcbcb"** are all happy strings and strings **"aa", "baa"** and **"ababbc"** are not happy strings.

Given two integers `n` and `k`, consider a list of all happy strings of length `n` sorted in lexicographical order.

Return *the kth string* of this list or return an **empty string** if there are less than `k` happy strings of length `n`.

Example 1:

```
Input: n = 1, k = 3
Output: "c"
Explanation: The list ["a", "b", "c"] contains all happy strings of length
1. The third string is "c".
```

Example 2:

```
Input: n = 1, k = 4
Output: ""
Explanation: There are only 3 happy strings of length 1.
```

Example 3:

```
Input: n = 3, k = 9
Output: "cab"
Explanation: There are 12 different happy string of length 3 ["aba", "abc",
"aca", "acb", "bab", "bac", "bca", "bcb", "cab", "cac", "cba", "cbc"]. You
will find the 9th string = "cab"
```

Example 4:

```
Input: n = 2, k = 7
Output: ""
```

Example 5:

```
Input: n = 10, k = 100
Output: "abacbabacb"
```

Constraints:

- `1 <= n <= 10`
- `1 <= k <= 100`

```
def getHappyString(self, n: int, k: int) -> str:
        if n==1 and k<4:
            return ['a','b','c'][k-1]
        ans = []
        ssf = ''
        self.solver(ans,ssf,n,k)
        return ans[k-1] if len(ans)>=k else ""


    def solver(self,ans,ssf,n,k):
        if n==0:
            ans.append(ssf)
            return
        for ele in ['a','b','c']:
            if len(ssf) != 0:
                if ssf[-1]!=ele:
                    self.solver(ans, ssf + ele, n - 1, k)
            else:
                self.solver(ans,ele,n-1,k)
```