

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

Problem Information

Difficulty: Medium

Acceptance Rate: 85.19%

Paid Only: No

Tags: String, Backtracking

Problem Description

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"

****Constraints:****

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

Code Snippets

C++:

```
class Solution {  
public:  
    string getHappyString(int n, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
public String getHappyString(int n, int k) {  
  
}  
}
```

Python3:

```
class Solution:  
    def getHappyString(self, n: int, k: int) -> str:
```