# 题目

给你一个二进制字符串 s 和一个整数 k 。如果所有长度为 k 的二进制字符串都是 s 的子串，请返回 true ，否则请返回 false 。

示例 1：

输入：s = "00110110", k = 2

输出：true

解释：长度为 2 的二进制串包括 "00"，"01"，"10" 和 "11"。它们分别是 s 中下标为 0，1，3，2 开始的长度为 2 的子串。

示例 2：

输入：s = "0110", k = 1

输出：true

解释：长度为 1 的二进制串包括 "0" 和 "1"，显然它们都是 s 的子串。

示例 3：

输入：s = "0110", k = 2

输出：false

解释：长度为 2 的二进制串 "00" 没有出现在 s 中。

提示：

1 <= s.length <= 5 \* 105

s[i] 不是'0' 就是 '1'

1 <= k <= 20

# 分析

## 方法一：哈希表

分析：

这里不要陷入，根据入参k去列举所有的二进制形式上面，主要是知道二进制数的数量，然后倒推。

代码：

class Solution {

public:

bool hasAllCodes(string s, int k) {

int n = s.length();

// 检查特殊情况，s 的长度小于 k，不可能包含所有长度为 k 的二进制串

if (n < k) {

return false;

}

unordered\_set<string> seen;

// 遍历字符串 s，并记录所有长度为 k 的二进制串

for (int i = 0; i <= n - k; ++i) {

string sub = s.substr(i, k);

seen.insert(sub);

}

// 如果哈希表中的元素数量等于 2^k，则说明所有长度为 k 的二进制串都存在

return seen.size() == (1 << k); // 检查哈希表中的元素数量是否等于 2^k

}

};