

重复至少 **K** 次且长度为 **M** 的模式

给你一个正整数数组 `arr`，请你找出一个长度为 `m` 且在数组中至少重复 `k` 次的模式。

模式 是由一个或多个值组成的子数组（连续的子序列），**连续** 重复多次但 **不重叠**。模式由其长度和重复次数定义。

如果数组中存在至少重复 `k` 次且长度为 `m` 的模式，则返回 `true`，否则返回 `false`。

示例 1：

输入：arr = [1,2,4,4,4,4], m = 1, k = 3

输出：true

解释：模式 **(4)** 的长度为 1，且连续重复 4 次。注意，模式可以重复 `k` 次或更多次，但不能少于 `k` 次。

示例 2：

输入：arr = [1,2,1,2,1,1,1,3], m = 2, k = 2

输出：true

解释：模式 **(1,2)** 长度为 2，且连续重复 2 次。另一个符合题意的模式是 **(2,1)**，同样重复 2 次。

示例 3：

输入：arr = [1,2,1,2,1,3], m = 2, k = 3

输出：false

解释：模式 **(1,2)** 长度为 2，但是只连续重复 2 次。不存在长度为 2 且至少重复 3 次的模式。

示例 4：

输入：arr = [1,2,3,1,2], m = 2, k = 2

输出：false

解释：模式 **(1,2)** 出现 2 次但并不连续，所以不能算作连续重复 2 次。

示例 5：

输入：arr = [2,2,2,2], m = 2, k = 3

输出: false

解释: 长度为 2 的模式只有 (2,2), 但是只连续重复 2 次。注意, 不能计算重叠的重复次数。

```
class Solution {
public:
    bool containsPattern(vector<int>& arr, int m, int k) {
        int len=arr.size();//统计源数组长度
        //由于模式只知道长度为 m, 具体模式的形式并不确定, 具体得看是否连续重复 k
        //次, 需要记录模式出现的次数
        int mod_times=0;
        //记录模式具体形式
        vector<int> temp_mod(m);
        //从源数组 从头开始向后遍历
        for(int i=0;i<len;i++)
        {
            if((len-i)<m*k)//剩余元素不够最小的匹配模式要求
                return false;
            for(int j=0;j<m;j++)//记录模式形式 以此为模板方便后续比对
            {
                temp_mod[j]=arr[i+j];
            }
            mod_times++;
            //下一个模式的开头和结尾 索引
            int next_start=i+m;
            int next_end=i+2*m-1;
            while(next_end<len)
            {
                int j;
                for(j=0;j<m;j++)
                {
                    if(temp_mod[j]!=arr[next_start+j])
                        break;
                }
                if(j==m)//此时说明匹配成功
                {
                    //下一匹配模式 首尾后移 m (m 为其模式长度)
                    next_start+=m;
                    next_end+=m;
                    mod_times++;
                    if(mod_times==k)//满足连续重复至少 k 次要求
                        return true;
                }
            }
        }
        else
    }
```

```
                break;//匹配不成功 跳出循环
            }
            //跳出循环后 此时认为该模式不为我们所要求的模式，重新需要找模式进行匹
配
            mod_times=0;
        }

        return false;//默认返回值为 false
    }
};
```