

剑指 Offer 61. 扑克牌中的顺子

Wednesday, June 22, 2022 10:09 AM

<https://leetcode.cn/problems/bu-ke-pai-zhong-de-shun-zi-lcof/>

简单

从若干副扑克牌中随机抽 5 张牌，判断是不是一个顺子，即这5张牌是不是连续的。

2~10为数字本身，A为1，J为11，Q为12，K为13，

而大、小王为 0，可以看成任意数字。

A 不能视为 14。

示例 2:

输入: [0,0,1,2,5]

输出: True

ps: A=1，和扑克不一样，10 J Q K A 不是顺子！

nums数组的值在 [0,13]

解法:

首先 顺子不能重复，用hashset判断重复hashMap也行。

其次 基于不重复的前提，若为顺子，其中最大值减去最小值 小于5。否则，顺不起来。

```
class Solution {
    public boolean isStraight(int[] nums) {
        Set<Integer> hset=new HashSet<>();
        // 默认最大值为0用于更新队内最大值，最小值设为13也是为了能更新最小值
        // 其实最大值初始为小于等于1的任何数都可，最小值设为大于等于13的任何数也行，无关紧要，只要能得到更新 反例，若min初始化比数组里任何一个数小，它就得不到更新
        int maxN=0,minN=13;
        for(int num : nums){
            // 0为大小王，可代替任何数，不用管，直接下一轮循环
            if(num==0){
                continue;
            }
            // 在hashSet中存在于了，重复 不是顺子，直接return
            if(hset.contains(num)){
                return false;
            }
            // 不存在 加入set
            hset.add(num);
            // 更新最大最小值
            maxN=Math.max(maxN,num);
            minN=Math.min(minN,num);
        }
    }
}
```

```
    }  
    return maxN-minN<5;  
  }  
}
```