剑指offer（python版）45-48

45扑克牌顺子

题目：LL今天心情特别好,因为他去买了一副扑克牌,发现里面居然有2个大王,2个小王(一副牌原本是54张^\_^)...他随机从中抽出了5张牌,想测测自己的手气,看看能不能抽到顺子,如果抽到的话,他决定去买体育彩票,嘿嘿！！“红心A,黑桃3,小王,大王,方片5”,“Oh My God!”不是顺子.....LL不高兴了,他想了想,决定大\小 王可以看成任何数字,并且A看作1,J为11,Q为12,K为13。上面的5张牌就可以变成“1,2,3,4,5”(大小王分别看作2和4),“So Lucky!”。LL决定去买体育彩票啦。 现在,要求你使用这幅牌模拟上面的过程,然后告诉我们LL的运气如何， 如果牌能组成顺子就输出true，否则就输出false。为了方便起见,你可以认为大小王是0

思路：所谓顺子只需要判断两点：1、五张牌没有重复 2、王牌个数等于顺子中空缺的位置。

代码：

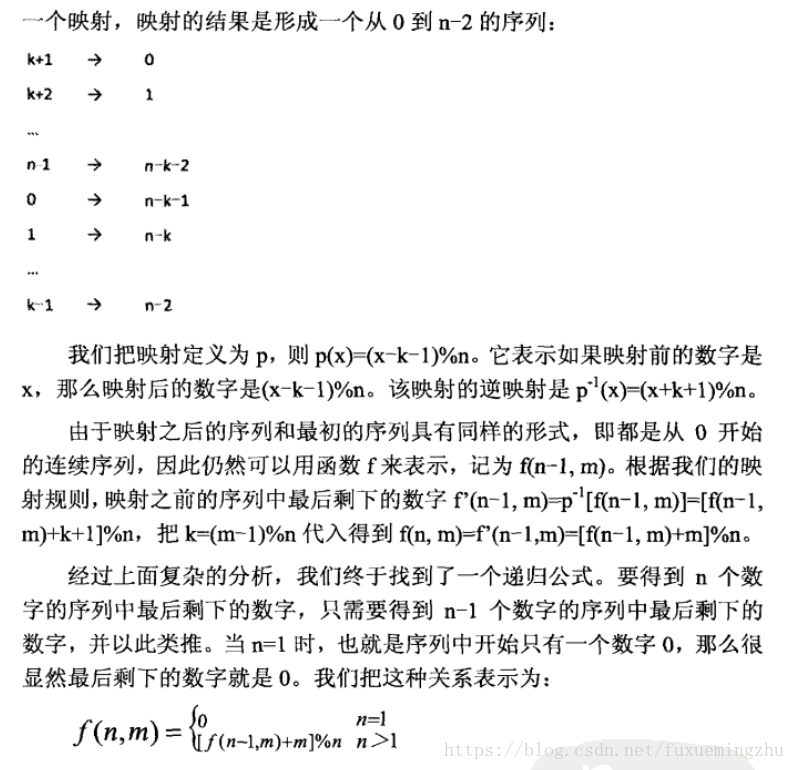
1. # -\*- coding:utf-8 -\*-
2. **class** Solution:
3. **def** IsContinuous(self, numbers):
4. # write code here
5. **if** len(numbers)==0:
6. **return** False
7. numbers.sort()
8. numofking=numbers.count(0)
9. numoflack=0
10. **for** i **in** range(numofking,len(numbers)-1):
11. **if** numbers[i]==numbers[i+1]:
12. **return** False
13. **if** numbers[i+1]==numbers[i]+1:
14. **continue**
15. **else**:
16. numoflack += (numbers[i+1]-numbers[i]-1)
17. **if** numoflack<=numofking:
18. **return** True
19. **else**:
20. **return** False

46孩子们的游戏(圆圈中最后剩下的数)

题目：每年六一儿童节,牛客都会准备一些小礼物去看望孤儿院的小朋友,今年亦是如此。HF作为牛客的资深元老,自然也准备了一些小游戏。其中,有个游戏是这样的:首先,让小朋友们围成一个大圈。然后,他随机指定一个数m,让编号为0的小朋友开始报数。每次喊到m-1的那个小朋友要出列唱首歌,然后可以在礼品箱中任意的挑选礼物,并且不再回到圈中,从他的下一个小朋友开始,继续0...m-1报数....这样下去....直到剩下最后一个小朋友,可以不用表演,并且拿到牛客名贵的“名侦探柯南”典藏版(名额有限哦!!^\_^)。请你试着想下,哪个小朋友会得到这份礼品呢？(注：小朋友的编号是从0到n-1)

思路：这道题即“约瑟夫环问题”。思路到位，代码异常简单。我阅读到的觉得不错的解释在

<https://www.cnblogs.com/cmmdc/p/7216726.html>



代码：

1. # -\*- coding:utf-8 -\*-
2. **class** Solution:
3. **def** LastRemaining\_Solution(self, n, m):
4. # write code here
5. **if** n<1 **or** m<1:
6. **return** -1
7. last=0
8. **for** i **in** range(2,n+1):
9. last=(last+m)%i
10. **return** last

47求1+2+3+...+n

题目：求1+2+3+...+n，要求不能使用乘除法、for、while、if、else、switch、case等关键字及条件判断语句（A?B:C）。

思路：递归 + 逻辑与的短路特性代替if语句

代码：

1. # -\*- coding:utf-8 -\*-
2. **class** Solution:
3. **def** \_\_init\_\_(self):
4. self.ans=0
5. **def** Sum\_Solution(self, n):
6. # write code here
7. self.recur(n)
8. **return** self.ans
9. **def** recur(self,n):
10. self.ans+=n
11. n-=1
12. **return** n>0 **and** self.Sum\_Solution(n)

48不用加减乘除做加法

题目：写一个函数，求两个整数之和，要求在函数体内不得使用+、-、\*、/四则运算符号。

思路：

第一步：两个数按位取亦或。获得的值为不考虑进位时的和。   
第二步：两个数按位取与。获得的值为（当两个数某一位同为1时）两个数求和时需要进位的位置。每次讲该数（在代码中表示为num2）左移一位，表示进位后。   
循环第一二步，直至两数的和没有进位位为止

32个1也就是一个int可表示的无符号整数为4294967295，对应的有符号为-1。因此最后我们可以判断符号位是否为1做处理。

https://www.jianshu.com/p/21fd1598d4ae

代码：

1. # -\*- coding:utf-8 -\*-
2. **class** Solution:
3. **def** Add(self, num1, num2):
4. # write code here
5. **while** num2 != 0:
6. temp = num1 ^ num2
7. num2 = (num1 & num2) << 1
8. num1 = temp & 0xFFFFFFFF
9. **return** num1 **if** num1 >> 31 == 0 **else** num1 - 4294967296