250 LuckyRemainder：给一个数X，定义super（X）为这个数的所有子集和，问super(x) %9 = ?。例如: Super(123) = 123 + 12 + 13 + 23 + 1 + 2 + 3 = 177。

答：如果要按照定义求Super(x)的话，可能会超时，注意到每一位都都可能是某一个某一个子集中的数字，我的想法就是求出每一个数字在不同子集中的权值和，然后在求和，这样对最后求和是没有影响的。于是我是这样计算的，假设X=x0x1x2..xi..xn,求xi的权值和，就需要计算出xi出现在每一个子集中的加权和，这样就可以分解成排列组合的问题。在进一步发现，不管是xi出现在什么位置，当前位置的基数%9都等于1（10 % 9 = 1， 100 % 9 = 1），所以也就相当于求xi出现的次数和了。在进一步发现，xi出现的次数与他左边和右边的数字有关，且次数为2 ^ (n – 1)。好了，最后的答案，就是求sum（dig）\* [2 ^ (n – 1)] % 9的值了。

500 PalindromizationDiv1：给一个字符串，允许在字符串上实施三种操作，删除，添加，更改，不过这三种操作都有不同的代价，问最少的代价将字符串改变成一个回文字符串。

答：如果没有三种操作代价相同，那么直接动态规划就可以了，但是三种代价不同，就略显得不同了，可能出现这么一个情况，删除一个字符串再添加一个字符串比更改一个字符串的代价要小，而且可能改变一个字符为一个字符被同时改变两个字符成另外一个字符的代价要答，所以要用floyd计算出更改的代价。然后就是动态规划了。