**19943:图的拉普拉斯矩阵**

def cs19943():  
 n, m = map(int, input().split())  
 a = [[0 for i in range(n)] for j in range(n)]  
 d = [0 for i in range(n)]  
 for i in range(m):  
 i1, i2 = map(int, input().split())  
 d[i1] += 1  
 d[i2] += 1  
 a[i1][i2] = 1  
 a[i2][i1] = 1  
 for i in range(n):  
 a[i][i] -= d[i]  
 for j in range(n - 1):  
 print(-a[i][j], end=" ")  
 print(-a[i][n - 1])



## 18160:最大连通域面积(matrix,dfs)

def cs18160():  
 def find(y, x):  
 if flag[y][x]:  
 flag[y][x] = False  
 tmp[0] += 1  
 for i in range(y - 1, y + 2):  
 for j in range(x - 1, x + 2):  
 if 0 <= i <= n - 1 and 0 <= j <= m - 1:  
 find(i, j)  
  
 for t in range(int(input())):  
 tmp, ans = [0], 0  
 n, m = map(int, input().split())  
 flag = [[True for j in range(m)] for i in range(n)]  
 ipt = []  
 for i in range(n):  
 j = 0  
 for js in input():  
 if js == ".":  
 flag[i][j] = False  
 j += 1  
 for i in range(n):  
 for j in range(m):  
 if flag[i][j]:  
 tmp[0] = 0  
 find(i, j)  
 ans = max(tmp[0], ans)  
 print(ans)



## 03441:4 Values whose Sum is 0

def cs03441():  
 a, b, c, d = [], [], [], []  
 dct, st1, st2 = {}, set(), set()  
 n = int(input())  
 for i in range(n):  
 ai, bi, ci, di = map(int, input().split())  
 a.append(ai)  
 b.append(bi)  
 c.append(ci)  
 d.append(di)  
 for i in list(a):  
 for j in list(b):  
 if i + j in st1:  
 dct[i + j] += 1  
 else:  
 dct[i + j] = 1  
 st1.add(i + j)  
 ans = 0  
 for i in list(c):  
 for j in list(d):  
 if -i - j in st1:  
 ans += dct[-i - j]  
 print(ans)



## 04089:电话号码

def cs04089():  
 for t in range(int(input())):  
 n = int(input())  
 pn = []  
 dct = {}  
 for i in range(n):  
 pn.append(input())  
 pn.sort(key=len, reverse=True)  
 flag = True  
 for i in range(n):  
 dct1 = dct  
 for j in pn[i][:-1]:  
 if not dct1.get(j):  
 dct1[j] = {-1: True}  
 dct1 = dct1[j]  
 if dct1.get(pn[i][-1]):  
 flag = False  
 break  
 else:  
 dct1[pn[i][-1]] = {-1: True}  
 if flag:  
 print("YES")  
 else:  
 print("NO")



## 04082:树的镜面映射

def cs04082():  
 def ad(knot):  
 if ipt[knot][1] == "0":  
 dct[knot] = [knot + 1]  
 tmp.append(knot)  
 else:  
 dct[knot] = [-1, -1]  
 if tmp:  
 dct[tmp.pop()].append(knot + 1)  
 if knot < n - 1:  
 ad(knot + 1)  
  
 def pp(knot):  
 if knot + 1:  
 dct[knot].append(l[0])  
 pp(dct[knot][1])  
 l[0] = dct[knot][2] + 1  
 mxl[0] = max(l[0], mxl[0])  
 pp(dct[knot][0])  
  
 l = [0]  
 mxl = [0]  
 tmp = []  
 n = int(input())  
 ipt = input().split()  
 dct = {}  
 ad(0)  
 pp(0)  
 ans, anss = [[] for i in range(mxl[0])], ""  
 for i in range(n):  
 ans[dct[i][2]].append(i)  
 for i in ans:  
 while i:  
 t = ipt[i.pop()][0]  
 if t != "$":  
 anss += t + " "  
 print(anss[:-1])



sw103383

def sw103383():  
 def cul(t):  
 if flag[t]:  
 tmp[0] += key[t]  
 flag[t] = False  
 for j in dct[t]:  
 cul(j)  
  
 n, m = map(int, input().split())  
 key = list(map(int, input().split()))  
 dct = {i: [] for i in range(n)}  
 flag = [True for i in range(n)]  
 ans, tmp = 0, [0]  
 for i in range(m):  
 a, b = map(int, input().split())  
 dct[a].append(b)  
 dct[b].append(a)  
 for i in range(n):  
 if flag[i]:  
 cul(i)  
 ans = max(ans, tmp[0])  
 tmp[0] = 0  
 print(ans)



总结：难度回升，总体可控。电话号码耗时较多，后面应用并查集解决。这两周进入期中季，我会减少花在数算上的时间