**Introduction to Computer Science**

**Homework 10**

**沙行勉教授**

**Due Date: 星期三，Dec. 19, 2018**

**Do not copy others, or you will get punished. If you allow other students to copy your report, you will be punished as well. In the homework, you will get more experience in programming. Good for you. 但是，对于编程题，我希望有基础的人能多帮助其他同学，帮助他们debug, 但是不要让他们抄袭。好吗？**

**PUT DOWN YOUR NAME and EMAIL ADDRESS IN YOUR REPORT.** The Filename of your report should be HOMEWORK N \_your Student ID\_your name \_Teacher’s last name (Sha or Zhang), where N is the homework number, for example, *1\_2018XXXX \_XiaoMing\_Sha*. Then send your report to the TA of your class. 注意，两班的TA不一样，千万不要送错了。**你可以用中文来回答问题。**

**请同学阅读书本的第五章。所有的编程都可以与一位同学合作，但是请写下合作同学的名字。**

1. **（每题8分）习题5.1， 5.3， 练习题5.5.1， 5.5.2， 5.6.1**

**习题5.1：**

**因为0！=1，1！=1，2！=1！\*2**

**所以可以得出 n!= 1(if n== 0 or n == 1)**

**n!= (n-1)!\*n**

**写成代码的话为：**

**def f(n):**

**if n == 0 or n == 1:**

**return 1**

**else:**

**return f(n-1)\*n**

**习题5.3：**

**a0=1, a1=1/(1+1)=1/2, a2=1/(1+a1)……**

**因此 an=1/(1+an-1)**

**练习题5.5.1：**

**T(0)=1**

**T(1)=T(0)=1**

**T(2)=T(1)+T(0)=2**

**先不管T（0），T(1),T(2)可以看成一个q=2的等比数列，T（3）=(1-4)/(1-2)+T(0)=4**

**正好是数列的新的项，T(4)=T(3)\*q-T(1)+T(0)=2[T(3)]**

**T(5)=T(3)\*q-T(1)+T(0)=2[T(4)]**

**T(6)=T(3)\*q-T(1)+T(0)=2[T(5)]**

**……因此推出从第二项开始T(n)=2[T(n-1)]**

**T(n)=2\*T(n-1)=2\*2\*T(n-2)=……=2\*\*(n-1)\*1=2n-1**

**证毕**

**练习题5.5.2**

**将x反转，逆向遍历，这样会先找到6**

**def lis(L):**

**Asc = [1]\*len(L)**

**Tra = [-1] \* len(L)**

**for i in range(1,len(L)):**

**x = []**

**for j in range(i):**

**if L[i] > L[j]:**

**x.append(j)**

**x.reverse()**

**for k in x:**

**if Asc[i] < Asc[k] + 1:**

**Asc[i] = Asc[k] + 1**

**Tra[i] = k**

**print("Asc:",Asc)**

**print("Tra:",Tra)**

**max = 0**

**for i in range(1,len(Asc)):**

**if Asc[i] > Asc [max]:**

**max = i**

**print("最长递增子序列的长度是",Asc[max])**

**x = [L[max]];i = max**

**while Tra[i] >= 0:**

**x=[L[Tra[i]]] + x**

**i = Tra[i]**

**print("最长递增子数列 =",x)**

**L = [5,2,4,7,6,3,8,9]**

**图片包含 屏幕截图

自动生成的说明lis(L)**

**练习题5.6.1**

**修改m后的输出结果：**

**图片包含 屏幕截图

自动生成的说明图片包含 屏幕截图

自动生成的说明调换起点和终点后的结果：**

1. **（15 points）编程， 习题5.4， 输出互质的分子和分母。也就是分子，分母**

**没有公因数。**

**def f(n):#求出分子式**

**if n == 0:**

**return 1,1**

**if n == 1:**

**return 1,2**

**return f(n-1)[1],(f(n-1)[0]+f(n-1)[1])**

**def ys(x,n):#看分子分母是否有最大公因数**

**if x<n:**

**x,n=n,x**

**while x%n!=0:**

**t=n**

**n=x%n**

**x=t**

**return n**

**n=int(input("请输入一个正整数:"))**

**a = f(n)[0];b = f(n)[1]**

**c = ys(a,b);a = a//c ;b = b//c**

**图片包含 屏幕截图

自动生成的说明print("an=%d/%d"%(a,b))**

1. **（15 points） 编程，习题5.7**

**习题5.7:**

**B = int(input("请输入您想要的进制的阿拉伯数字:"))**

**L1 = list(map(int,input().split()))#按每一位输入，例如16进制的1099，请输入 10 9 9，数字之间以空格分隔**

**L2 = list(map(int,input().split()))#同上**

**def add\_B(L1,L2,c=False):**

**if len(L1) > len(L2):**

**L2 = [0]\*(len(L1)-len(L2))+L2**

**else:**

**L1 = [0]\*(len(L2)-len(L1))+L1**

**if len(L1) == 1:**

**re = L1[-1] + L2[-1]**

**if c == True:**

**re+=1**

**if re < B:**

**return [re]**

**else:**

**return [1]+[re-B]**

**re = L1[-1] + L2[-1]**

**if c == True:**

**re += 1**

**if re >= B:**

**return add\_B(L1[:len(L1)-1],L2[:len(L2)-1],c=True)+[re-B]**

**else:**

**return add\_B(L1[:len(L1)-1],L2[:len(L2)-1],c=False)+[re]**

**print(add\_B(L1,L2))**

**图片包含 屏幕截图

自动生成的说明图片包含 屏幕截图

自动生成的说明**

1. **（10 points）编程， 习题5.9， 输出一个列表。例如，输入整数2351554， 返回一个列表 [2,3,5,1,5,5,4]**

**习题5.9**

**def f(s):**

**if len(s) == 1:**

**return [int(s[0])]**

**return f(s[0:len(s)-1]) + f(s[-1])**

**x = input()**

**图片包含 屏幕截图

自动生成的说明print(f(x))**

1. **（15 points）编程，习题5.14, 请打印n从20到60的个数。你会发现跑到n=35 左右，你的电脑可能就跑不完了。请改写你的程序，让它能飞快地完成。**

**习题5.14**

**#从n大于3开始，因为n阶台阶的走法的最后一步一定是1步或者2步或者3步，所以**

**#f(n)=f(n-1)+f(n-2)+f(n-3)**

**a = [1,2,4005D**

**for i in range(3,60):**

**a.append(a[i-1]+a[i-2]+a[i-3])**

**for i in range(19,60):**

**图片包含 屏幕截图

自动生成的说明print("f(%d)=%d"%(i+1,a[i]),end=" ")**

1. **（5 points）请简单写下你们团队关于期末大项目的情形。  
   你们最后要做展示，预计是1月10号，交上电子档的代码和报告。你们团队合作，有队友可能分工写界面代码，有队友写报告，注意最终报告内容需要写游戏使用手册，程序发展过程，基本思路，分工情形，将来的扩展工作等。你们的程序将展示给大家（老师，同学，学长们及将来的学弟妹），所以希望大家好好表现。**

**我们经过反复协商，打算在我们的能力范围内制作一款原理简单但是趣味性高的游戏，于是我们打算制作2048,也就是一款数字消除游戏，所要用的算法涉及到子列表和矩阵，同时我们会尝试加入pygame来提升观赏性和可玩性，目前我们的算法已经有了逐步拟定，并打算近期完成一个基本雏形**