HW#D

经济学院 刘安澜 1700015495

OJ2981: 大整数加法（送分题目）  
<http://cs101.openjudge.cn/practice/2981/>

解题思路：这题确实是送分题🤣🤣🤣。

1. ﻿﻿m=int(input())
2. n=int(input())
3. **print**(m+n)

OJ18161: 矩阵运算(matrix)，cs101-2017 期末机考备选  
http://cs101.openjudge.cn/practice/18161/

解题思路：因为学过线性代数，所以对这道题的运算比较熟悉，也能很快写出判断条件。

1. rowA,colA=map(int,input().split())
2. A=[[int(x) **for** x **in** input().split()] **for** i **in** range(rowA)]
3. rowB,colB=map(int,input().split())
4. B=[[int(x) **for** x **in** input().split()] **for** i **in** range(rowB)]
5. rowC,colC=map(int,input().split())
6. C=[[int(x) **for** x **in** input().split()] **for** i **in** range(rowC)]
7. **if** colA!=rowB **or** rowA!=rowC **or** colB!=colC:
8. **print**('Error!')
9. **else**:
10. ans=[[0]\*colB **for** i **in** range(rowA)]
11. **for** i **in** range(rowA):
12. **for** j **in** range(colB):
13. **for** k **in** range(colA):
14. ans[i][j]+=A[i][k]\*B[k][j]
15. ans[i][j]+=C[i][j]
16. **for** i **in** range(rowC):
17. **print**(' '.join([str(x) **for** x **in** ans[i]]))

OJ12559: 最大最小整数 v0.2(string,sorting)，cs101 2016 final exam  
http://cs101.openjudge.cn/practice/12559/

解题思路：是这次作业中最没有思路的题目，一开始认为是简单的字符串排序，后来发现对于9和9123会给出错误的排序。后来查了冒泡排序，但还是看到了群里同学提供的冒泡排序筛选思路才明白了（否则暴力循环一定会超时），这样时间复杂度可以迅速降低。这个排序方式非常巧妙，因为两个数前后组合的位数是相等的，就不会出现不同长度字符串排序的干扰。

1. **def** bubble\_sort(array):
2. **for** i **in** range(1, len(array)):
3. **for** j **in** range(0, len(array)-i):
4. **if** array[j]+array[j+1] > array[j+1]+array[j]:
5. array[j], array[j+1] = array[j+1], array[j]
6. **return** array
8. n=int(input())
9. l=input().split()
10. sortl=bubble\_sort(l)
11. **print**(''.join([str(x) **for** x **in** reversed(sortl)]),''.join([str(x) **for** x **in** sortl]))

OJ19963: 买学区房v0.3(sort,math), cs101 2019 Final Exam  
http://cs101.openjudge.cn/practice/19963/

解题思路：感觉是一道比较中规中矩的排序题目，难点主要在坐标的处理上，我先把字符串两头去掉再按照括号分割，之后取数字即可。我觉得自己的写法开了很多列表，其实有些冗余，但好在本题样本量不大。中位数的提取，因为不能使用numpy库，所以只能相对麻烦写一个判断。

PS：这题最开始困惑我的是，房子距离越近，价格越低对购买者越有利，但是为啥性价比是越高越好（？？？？），这样房子距离越远价格越低性价比越高……感觉这个经济意义有点问题😅

1. n=int(input())
2. l=input()
3. price=[int(x) **for** x **in** input().split()]
4. l1=l[1:len(l)-2].split(') (')
5. l2=[[0,0] **for** i **in** range(n)]
6. distance=[0]\*n
7. ratio=[0]\*n
8. **for** i **in** range(n):
9. l2[i][0],l2[i][1]=map(int,l1[i].split(','))
10. distance[i]=l2[i][0]+l2[i][1]
11. ratio[i]=distance[i]/price[i]
12. **if** n%2==0:
13. ratiom=(sorted(ratio)[n//2]+sorted(ratio)[n//2-1])/2
14. pricem=(sorted(price)[n//2]+sorted(price)[n//2-1])/2
15. **else**:
16. ratiom=sorted(ratio)[n//2]
17. pricem=sorted(price)[n//2]
18. ans=0
19. **for** i **in** range(n):
20. **if** price[i]<pricem **and** ratio[i]>ratiom:
21. ans+=1
22. **print**(ans)

OJ19948: 因材施教 (greedy), cs101 2019 Final Exam  
http://cs101.openjudge.cn/practice/19948/

解题思路：这题一开始没有太多思路，能想到的就是要开一个差值的列表。但后来忽然明白本题的本质就是去掉m-1个最大的差值，所以最后的程序很简单。感觉这种题目思路通了就非常顺了。

1. n,m=map(int,input().split())
2. l=sorted([int(x) **for** x **in** input().split()])
3. diff=[0]\*(n-1)
4. **for** i **in** range(n-1):
5. diff[i]=l[i+1]-l[i]
6. **print**(sum(sorted(diff,reverse=True)[m-1:]))

//选做：

OJ4101: 晶矿的个数 (dfs)  
http://cs101.openjudge.cn/practice/4101/

解题思路：两道dfs题目我都是直接借用上次池塘题目的代码改的，感觉这类题目非常类似，所以也很好借鉴。这道题看了一下答案，我觉得我写的有些麻烦，应该在dfs函数里面再加一个晶矿类型的参数就更方便。

1. dx=[0,0,1,-1]
2. dy=[1,-1,0,0]
3. **def** dfsr(i,j):
4. **if** l[i][j]!='r':
5. **return**
6. l[i][j]='R'
7. **for** s **in** range(4):
8. dfsr(i+dx[s],j+dy[s])
10. **def** dfsb(i,j):
11. **if** l[i][j]!='b':
12. **return**
13. l[i][j]='B'
14. **for** s **in** range(4):
15. dfsb(i+dx[s],j+dy[s])
17. T=int(input())
18. **for** i **in** range(T):
19. N=int(input())
20. l=[[0]\*(N+2)]+[[0]+list(input())+[0] **for** i **in** range(N)]+[[0]\*(N+2)]
21. countr=0
22. countb=0
23. **for** i **in** range(1,N+1):
24. **for** j **in** range(1,N+1):
25. **if** l[i][j]=='r':
26. countr+=1
27. dfsr(i,j)
28. **if** l[i][j]=='b':
29. countb+=1
30. dfsb(i,j)
31. **print**(countr,countb)

OJ18160: 最大连通面积 (dfs)，cs101-2017 期末机考备选  
http://cs101.openjudge.cn/practice/18160/

解题思路：代码整体上还是参考池塘的dfs。本题是我第一次使用全局变量声明，所以对这类需要在循环内调用外部的变量的形式有了更多的认识。不过貌似OJ平台上对于先函数定义的global变量不友好，必须在出现之前先赋值一下变量才能过……

<https://blog.csdn.net/qq_28888837/article/details/88060376>

1. dx=[0,0,1,1,1,-1,-1,-1]
2. dy=[1,-1,1,0,-1,1,0,-1]
3. count=0
4. **def** dfs(i,j):
5. **global** count
6. **if** l[i][j]!='W':
7. **return**
8. l[i][j]='M'
9. count+=1
10. **for** s **in** range(8):
11. dfs(i+dx[s],j+dy[s])
13. T=int(input())
14. **for** i **in** range(T):
15. N,M=map(int,input().split())
16. l=[[0]\*(M+2)]+[[0]+list(input())+[0] **for** i **in** range(N)]+[[0]\*(M+2)]
17. ans=0
18. **for** i **in** range(1,N+1):
19. **for** j **in** range(1,M+1):
20. count=0
21. **if** l[i][j]=='W':
22. dfs(i,j)
23. ans=max(ans,count)
24. **print**(ans)

OJ21608: 你和你比较熟悉的同学 (dfs, bfs)，2020fall 沉浸式教学样例

http://cs101.openjudge.cn/practice/21608/

解题思路：这题我参考了老师提供的dfs思路，在自己复现的时候又改进了一下细节：一个是如果节点为-1代表这个节点后面没有数据，所以直接在函数内处理一下判断条件；还有一个是按照' : '分割可以去掉id后面的空格，就不需要再使用rstrip函数了。本题的关键是首先生成一个完整的图，然后对图里的元素进行遍历索引，标记索引过的元素。个人认为这道题dfs思路比bfs更加容易明白，这道题算是提供了一个对于图类问题的解决模板。

1. **def** dfs(graph,node,visited):
2. **if** node!=-1 **and** node **not** **in** visited:
3. visited.append(node)
4. **if** node **not** **in** graph:
5. **return** visited
6. **for** nei **in** graph[node]:
7. dfs(graph,nei,visited)
8. **return** visited
10. graph={}
11. ids=set()
12. n=int(input())
13. **for** i **in** range(n):
14. l=input().split(' : ')
15. ids.add(int(l[0]))
16. **if** int(l[0]) **not** **in** graph:
17. graph[int(l[0])]=[int(x) **for** x **in** l[1].split()]
19. maxp=0
20. **for** i **in** ids:
21. dfs\_path=dfs(graph,i,[])
22. maxp=max(maxp,len(dfs\_path))
23. **print**(maxp)

PS：这学期学完计概之后，本周在写MATLAB的时候也将计概的一些循环技巧套用在了里面，觉得自己编程水平确实提高了🤣还是挺有收获的！