

Assignment #9: Mock Exam立冬前一天

Updated 1658 GMT+8 Nov 6, 2025

2025 fall, Compiled by 郑文萱 基础医学院

说明:

1. Nov月考: AC3 (请改为同学的通过数)。考试题目都在“题库 (包括计概、数算题目)”里面, 按照数字题号能找到, 可以重新提交。作业中提交自己最满意版本的代码和截图。
2. 解题与记录: 对于每一个题目, 请提供其解题思路 (可选), 并附上使用Python或C++编写的源代码 (确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted)。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。(推荐使用Typora <https://typoraio.cn> 进行编辑, 当然你也可以选择Word。)无论题目是否已通过, 请标明每个题目大致花费的时间。
3. 提交安排: 提交时, 请首先上传PDF格式的文件, 并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像, 提交的文件为PDF格式, 并且“作业评论”区包含上传的.md或.doc附件。
4. 延迟提交: 如果你预计无法在截止日期前提交作业, 请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业, 以保证顺利完成课程要求。

1. 题目

E29982:一种等价类划分问题

hashing, <http://cs101.openjudge.cn/practice/29982>

思路: 此题看起来简单但坑点满满: 本人第一次错是循环语句的顺序没搞对, 导致只能记录第一个数; 第二次错是因为没有将列表转换为字符串再加逗号, 语法必须要过关啊。

代码

```
m,n,k=map(int,input().split(','))
bag={}
for i in range(m+1,n):
    sumnum=sum(int(n) for n in str(i))
    if sumnum%k==0:
        if sumnum not in bag:
            bag[sumnum]=[]
        bag[sumnum].append(i)
for sumnum in sorted(bag.keys()):
    sumnum=sorted(bag[sumnum])
    print(','.join(map(str,sumnum)))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
m,n,k=map(int,input().split(','))
bag={}
for i in range(m+1,n):
    sumnum=sum(int(n) for n in str(i))
    if sumnum%k==0:
        if sumnum not in bag:
            bag[sumnum]=[]
        bag[sumnum].append(i)
for sumnum in sorted(bag.keys()):
    sumnum=sorted(bag[sumnum])
    print(' '.join(map(str,sumnum)))
```

基本信息

#: 50755051
题目: 29982
提交人: 25n2510305212(zx)
内存: 4160kB
时间: 25ms
语言: Python3
提交时间: 2025-11-08 17:22:08

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

E30086:dance

greedy, <http://cs101.openjudge.cn/practice/30086>

思路: 这个瞪眼法可知是相邻比较, 不过我写循环经常容易出小问题, 特别是要加条件的, 这个WA了3次后才过, 碰到样例少的题一直WA真的会没招, 要学Debug了呜呜呜。

代码

```
n,d=map(int,input().split())
students=sorted(list(map(int,input().split())))
found=True
for i in range(0,2*n,2):
    if students[i+1]-students[i]>d:
        found=False
        break
print('Yes' if found else 'No')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n,d=map(int,input().split())
students=sorted(list(map(int,input().split())))
found=True
for i in range(0,2*n,2):
    if students[i+1]-students[i]>d:
        found=False
        break
print('Yes' if found else 'No')
```

©2002-2022 POJ 京ICP备20010980号-1

M25570: 洋葱

matrices, <http://cs101.openjudge.cn/practice/25570>

思路: 这个就很经典的矩阵题目, 但我第一次WA了, 主要因为忽略了奇数最中间只有一个, 重复算了两遍, 加了条件后AC。看了题解之后发现了更好的办法, 不用暴力写循环。

代码

```
n=int(input())
onion=[]
layer=(n+1)//2
maxsum=0
for i in range(n):
    onion.append(list(map(int,input().split())))
for k in range(layer):
    count=0
    for i in range(k,n-k):
        count+=onion[k][i]
    if k!=n-k-1:
        for i in range(k,n-k):
            count+=onion[n-k-1][i]
    for j in range(k+1,n-k-1):
        count+=onion[j][k]
    for j in range(k+1,n-k-1):
        count+=onion[j][n-k-1]
    if count>maxsum:
        maxsum=count
print(maxsum)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n=int(input())
onion=[]
layer=(n+1)//2
maxsum=0
for i in range(n):
    onion.append(list(map(int,input().split())))
for k in range(layer):
    count=0
    for i in range(k,n-k):
        count+=onion[k][i]
    if k!=n-k-1:
        for i in range(k,n-k):
            count+=onion[n-k-1][i]
    for j in range(k+1,n-k-1):
        count+=onion[j][k]
    for j in range(k+1,n-k-1):
        count+=onion[j][n-k-1]
    if count>maxsum:
        maxsum=count
print(maxsum)
```

©2002-2022 POJ 京ICP备20010980号-1

M28906:数的划分

dfs, dp, <http://cs101.openjudge.cn/practice/28906>

思路: 此题是计时结束后AC的, 对dfs不是很熟练, 所以限定时间内磨了半天无成果。结束后学习了dp的做法, 很简洁很优雅, 难度比较大的部分是独立推出表达式, 老师快点讲讲dp吧。

代码

```
n,k=map(int,input().split())
dp=[[0]*(k+1)for _ in range(n+1)]
dp[0][0]=1
for i in range(1,n+1):
    for j in range(1,k+1):
        if i>=j:
            dp[i][j]= dp[i-1][j-1]+dp[i-j][j]
print(dp[n][k])
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n, k = map(int, input().split())
dp = [[0] * (k + 1) for _ in range(n + 1)]
dp[0][0] = 1
for i in range(1, n + 1):
    for j in range(1, k + 1):
        if i >= j:
            dp[i][j] = dp[i - 1][j - 1] + dp[i - j][j]
print(dp[n][k])
```

©2002-2022 POJ 京ICP备20010980号-1

M29896:购物

greedy, <http://cs101.openjudge.cn/practice/29896>

思路: 这个题是很明显的贪心加dp (话说为什么月考就考dp啊根本不会), 计时时间结束之后做的, 一开始的思路是: 新覆盖的范围=已覆盖的范围+新硬币面值 (也可以说是已覆盖的范围), 但用dp不停WA, 最后参考了题解后完全理解, 只用了贪心。如果已经能凑出X, 那么接下来的X+A以内都可以凑出, A为小于X的最大硬币面值, 这个原理我花了两天才想明白。

代码

```
from bisect import bisect_right

def solve():
    X, N = map(int, input().split())
    coins = list(map(int, input().split()))
    coins.sort()
    if coins[0] > 1:
        print(-1)
        return
    cover = 0
    count = 0
    while cover < X:
        target = cover + 1
        idx = bisect_right(coins, target) - 1
        if idx < 0:
            print(-1)
            return
        cover += coins[idx]
        count += 1
    print(count)
solve()
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from bisect import bisect_right

def solve():
    X, N = map(int, input().split())
    coins = list(map(int, input().split()))
    coins.sort()
    if coins[0] > 1:
        print(-1)
        return
    cover = 0
    count = 0
    while cover < X:
        target = cover + 1
        idx = bisect_right(coins, target) - 1
        if idx < 0:
            print(-1)
            return
        cover += coins[idx]
        count += 1
    print(count)

solve()
```

©2002-2022 POJ 京ICP备20010980号-1

T25353:排队

greedy, <http://cs101.openjudge.cn/practice/25353>

思路: 这个题看了之后其实不太有思路, 题解也看得一头雾水, 与D老师交流后明白了题解的做法, 非常明确的思路, 就是分组然后看能不能加入当前组, 然后组内可以排序, 又因为组间不能交换, 所以这就是正确做法。

代码

```
N, D = map(int, input().split())
height = [0] * N
judge = [False] * N
for i in range(N):
    height[i] = int(input())
output = []
while False in judge:
    i, r = 0, len(height)
    now_list = []
    while i < r:
        if judge[i]:
```

```
        i+=1
        continue
    if len(now_list)==0:
        now_list.append(height[i])
        maxh,minh=height[i],height[i]
        judge[i]=True
        continue
    maxh=max(height[i],maxh)
    minh=min(height[i],minh)
    if maxh-height[i]<=D and height[i]-minh<=D:
        now_list.append(height[i])
        judge[i]=True
    i+=1
now_list.sort()
output.extend(now_list)
print(*output,sep='\n')
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
N,D=map(int,input().split())
height=[0]*N
judge=[False]*N
for i in range(N):
    height[i]=int(input())
output=[]
while False in judge:
    i,r=0,len(height)
    now_list=[]
    while i<r:
        if judge[i]:
            i+=1
            continue
        if len(now_list)==0:
            now_list.append(height[i])
            maxh,minh=height[i],height[i]
            judge[i]=True
            continue
        maxh=max(height[i],maxh)
        minh=min(height[i],minh)
        if maxh-height[i]<=D and height[i]-minh<=D:
            now_list.append(height[i])
            judge[i]=True
        i+=1
    now_list.sort()
    output.extend(now_list)
print(*output,sep='\n')
```

©2002-2022 POJ 京ICP备20010980号-1

2. 学习总结和收获

期中考完终于满血复活了，这个是自己计时做的，只做出前三道，发现期中那两周讲的内容掌握的不是很好，在补了，每日选做也在补了。这次题出的很综合，属于全面考察了已学知识（以及未学知识）：哈希表、矩阵、贪心、递归、dp。语法上学到了bisect用于查找索引，sep='?'用于在参数间加分隔符。现在离独立做出后两道这种难题还差得很远，路漫漫啊。