

Assignment #B: Dec Mock Exam 大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</mark>

说明：

- 1) 月考：AC2<mark>（请改为同学的通过数）</mark>。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) 请把每个题目解题思路（可选），源码 Python, 或者 C++（已经在 Codeforces/Openjudge 上 AC），截图（包含 Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用 word）。AC 或者没有 AC，都请标上每个题目大致花费时间。
- 3) 提交时候先提交 pdf 文件，再把 md 或者 doc 文件上传到右侧“作业评论”。Canvas 需要有同学清晰头像、提交文件有 pdf、“作业评论”区有上传的 md 或者 doc 附件。
- 4) 如果不能在截止前提交作业，请写明原因。

1. 题目

E22548: 机智的股民老张

<http://cs101.openjudge.cn/practice/22548/>

思路：

（考场原代码）天崩开局 TLE，当时太慌了没想出来啥好办法，就用了 dilworth 定理找最长上升子链的过程去更新最大差值，，，考场上真的被第一题搞懵了。。。。（我同学说我这是在用大炮打蚊子哈哈哈）

代码：

```
from bisect import bisect_right
s=list(map(int,input().split()))
ans=0
lis=[]
for i in s:
    pos=bisect_right(lis,i)
    if pos<len(lis):
        lis[pos]=i
    else:
        lis.append(i)
    ans=max(ans,lis[-1]-lis[0])
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
from bisect import bisect_right
s=list(map(int,input().split()))
ans=0
lis=[]
for i in s:
    pos=bisect_right(lis,i)
    if pos<len(lis):
        lis[pos]=i
    else:
        lis.append(i)
    ans=max(ans,lis[-1]-lis[0])
print(ans)
```

基本信息

#: 47572838
题目: 22548
提交人: 24n2400016635
内存: 9508kB
时间: 58ms
语言: Python3
提交时间: 2024-12-05 17:40:10

©2007-2022 P01 吉ICP备20010980号-1

English 帮助

M28701: 炸鸡排

greedy, <http://cs101.openjudge.cn/practice/28701/>

思路:

(考后看的题解)我觉得这种 对每个时间进行平均值的比较来分割或删除 非常不显然, 非常不好猜, 我即使去洗了澡吃了好吃的也没有自己灵光一现想出来。看题解可以看懂, 自己想不出来。

题解代码是要首先注意到最长时间是所有时间总和 对 k 的平均值, 相当于把所有时间拼在一起再剪成 k 段, 这样一直都能利用锅了。(我考试以及考试结束自己再想, 都没注意到这个)。接着, 由于剪成的 k 段要同时放到锅里, 因此不能出现分割了的时间段同时出现在锅里面, 而这就是时间段长于上面的平均值的, 因此要把这个删掉, 自己成为一个段, 将剩下的 $n-1$ 个时间段再拼接并剪成 $k-1$ 段。这样做的前提是题目说了每一个鸡排都可以随时取出。

代码:

```
n,k=map(int,input().split())
s=sorted(list(map(int,input().split()))),reverse=True)
total=sum(s)
for i in range(n):
    if s[i]<=total/k:
        break
    else:
        total-=s[i]
        k-=1
print(format(eval('total/k'),'.3f'))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
n,k=map(int,input().split())
s=sorted(list(map(int,input().split())),reverse=True)
total=sum(s)
for i in range(n):
    if s[i]<=total/k:
        break
    else:
        total-=s[i]
        k-=1
print(format(eval('total/k'),'.3f'))
```

基本信息

#: 47617852
题目: 28701
提交人: 24n2400016635
内存: 3608kB
时间: 21ms
语言: Python3
提交时间: 2024-12-08 00:27:07

M20744: 土豪购物

dp, <http://cs101.openjudge.cn/practice/20744/>

思路:

(考后看的题解) 考场上有想过从取和不取的思路,但是没想出来转移方程和 dp 的定义问题。这里妙的是 $dp[i][0]$ 是题目所求, $dp[i][1]$ 更像是一个辅助 dp, 因为求的是不存在放回时的情况, 所以最后 max 取的是 $dp[i][0]$ 。此外初始化都要设置成 $s[0]$ 是为了防止都为负数的情况, 因为如果设置成 $\max(0, s[0])$ 的话会出错。

代码:

```
s=list(map(int,input().split(',')))
#dp:取? 不取?
#dp[i][0]表示到第 i 个时, 有可能已经出现了不取的情况->该处不取? 之前有可能已有不取的该处取? 该处单独取比之前还大?
#dp[i][1]表示到第 i 个时, 未出现不取的情况->接着取该处? 该处单独取比之前的都大?
dp=[[0,0] for _ in range(len(s))]
dp[0][0]=s[0]
dp[0][1]=s[0]
for i in range(1,len(s)):
    dp[i][0]=max(dp[i-1][1],max(dp[i-1][0]+s[i],s[i]))
    dp[i][1]=max(dp[i-1][1]+s[i],s[i])
print(max(dp[i][0] for i in range(len(s))))
```

代码运行截图 <mark> (至少包含有"Accepted") </mark>

状态: Accepted

源代码

```
s=list(map(int,input().split(',')))
#dp:取? 不取?
#dp[i][0]表示到第i个时,有可能已经出现了不取的情况->该处不取?之前有可能已有不取的情况
#dp[i][1]表示到第i个时,未出现不取的情况->接着取该处?该处单独取比之前的都大?
dp=[ [0,0] for _ in range(len(s)) ]
dp[0][0]=s[0]
dp[0][1]=s[0]
for i in range(1,len(s)):
    dp[i][0]=max(dp[i-1][1],max(dp[i-1][0]+s[i],s[i]))
    dp[i][1]=max(dp[i-1][1]+s[i],s[i])
print(max(dp[i][0] for i in range(len(s))))
```

基本信息

#: 47618052
题目: 20744
提交人: 24n2400016635
内存: 13008kB
时间: 89ms
语言: Python3
提交时间: 2024-12-08 01:03:41

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

T25561: 2022 决战双十一

brute force, dfs, <http://cs101.openjudge.cn/practice/25561/>

思路:

(考完自己做出来的)我觉得最重要的是先设置好参数和参数的意义,再调整输入的格式。我设置的参数是: k---int---第 k 个商品; shop---list---存储当前在各个商店的金额数; curr1---int---各个商店打折后的价格总额; curr2---int---各个商店没有打折时的价格总额。

代码:

```
def getMaxBenefit(cut,v):
    c=0
    for i in range(len(cut)):
        if cut[i][0]>v:
            break
        c=max(c,cut[i][1])
    return c

ans=[]
def dfs(k,shop,curr1,curr2):
    global ans
    if k==n:
        for i in range(m):
            v=shop[i]
            curr2+=v
            real_v=v-getMaxBenefit(cut[i],v)
            curr1+=real_v
            curr1-=50*(curr2//300)
            ans.append(curr1)
            curr1=curr2=0
        return

    for i in range(len(goods[k])):
        shop[goods[k][i][0]-1]=shop[goods[k][i][0]-1]+goods[k][i][1]
        dfs(k+1,shop,curr1,curr2)
        shop[goods[k][i][0]-1]=shop[goods[k][i][0]-1]-goods[k][i][1]
```

```
n,m=map(int,input().split())
goods=[[] for _ in range(n)]
shop=[0]*m
cut=[[] for _ in range(m)]
for i in range(n):
    s=list(input().split())
    for j in s:
        a,b=map(int,j.split(':'))
        goods[i].append((a,b))
for i in range(m):
    s=list(input().split())
    for j in s:
        a,b=map(int,j.split('-'))
        cut[i].append((a,b))
    cut[i].sort(key=lambda x:x[0])
dfs(0,shop,0,0)
print(min(ans))
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

状态: Accepted

基本信息

#: 47584418
题目: 25561
提交人: 24n2400016635
内存: 3736kB
时间: 40ms
语言: Python3
提交时间: 2024-12-06 14:55:58

源代码

```
def getMaxBenefit(cut,v):
    c=0
    for i in range(len(cut)):
        if cut[i][0]>v:
            break
        c=max(c,cut[i][1])
    return c

ans=[]
def dfs(k,shop,curr1,curr2):
    global ans
    if k==n:
        for i in range(m):
            v=shop[i]
            curr2+=v
            real_v=getMaxBenefit(cut[i],v)
            curr1+=real_v
            curr1-=50*(curr2//300)
            ans.append(curr1)
            curr1=curr2=0
        return

    for i in range(len(goods[k])):
        shop[goods[k][i][0]-1]=shop[goods[k][i][0]-1]+goods[k][i][1]
        dfs(k+1,shop,curr1,curr2)
        shop[goods[k][i][0]-1]=shop[goods[k][i][0]-1]-goods[k][i][1]

n,m=map(int,input().split())
goods=[]
for _ in range(n):
    shop=[0]*m
    cut=[]
    for _ in range(m):
        for i in range(n):
            s=list(input().split())
            for j in s:
                a,b=map(int,j.split(':'))
                goods[i].append((a,b))
        for i in range(m):
            s=list(input().split())
            for j in s:
                a,b=map(int,j.split('-'))
                cut[i].append((a,b))
        cut[i].sort(key=lambda x:x[0])
    dfs(0,shop,0,0)
print(min(ans))
```

T20741: 两座孤岛最短距离

dfs, bfs, <http://cs101.openjudge.cn/practice/20741/>

思路:

(考场原代码) 思路是第一次 bfs 找到第一个孤岛的所有坐标, 第二次 bfs 以第一个孤岛的所有坐标为起点, 设置 step, 拓展到第二个孤岛就可以 return step。虽然代码有两个 bfs 代码, 但是手搓一个之后再复制粘贴修改就好了。令我惊讶的是考场上这道题居然一次提交就 ac 了, 本来担心超时的。

代码:

```
from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x,y):
    start=deque()
    q=deque()
```

```

        q.append((x,y))
        start.append((x,y))
        inq=set()
        inq.add((x,y))
        while q:
            for _ in range(len(q)):
                x,y=q.popleft()
                for i in range(4):
                    nx,ny=x+dx[i],y+dy[i]
                    if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and
s[nx][ny]=='1':
                        q.append((nx,ny))
                        start.append((nx,ny))
                        inq.add((nx,ny))

        return start
def bfs2(start):
    inq=set(start)
    step=0
    while start:
        step+=1
        for _ in range(len(start)):
            x,y=start.popleft()
            for i in range(4):
                nx,ny=x+dx[i],y+dy[i]
                if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and
s[nx][ny]=='1':
                    return step-1
                if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and
s[nx][ny]=='0':
                    start.append((nx,ny))
                    inq.add((nx,ny))

n=int(input())
s=[input() for _ in range(n)]
m=len(s[0])
f=0
for i in range(n):
    if f:
        break
    for j in range(m):
        if s[i][j]=='1':
            f=1
            start=bfs(i,j)
            break

step=bfs2(start)
print(step)

```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

状态: Accepted

源代码

```
from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x,y):
    start=deque()
    q=deque()
    q.append((x,y))
    start.append((x,y))
    inq=set()
    inq.add((x,y))
    while q:
        for _ in range(len(q)):
            x,y=q.popleft()
            for i in range(4):
                nx,ny=x+dx[i],y+dy[i]
                if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and s[nx]
                    q.append((nx,ny))
                    start.append((nx,ny))
                    inq.add((nx,ny))
    return start
def bfs2(start):
    inq=set(start)
    step=0
    while start:
        step+=1
        for _ in range(len(start)):
            x,y=start.popleft()
            for i in range(4):
                nx,ny=x+dx[i],y+dy[i]
                if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and s[nx]
                    return step-1
                if 0<=nx<n and 0<=ny<m and (nx,ny) not in inq and s[nx]
                    start.append((nx,ny))
                    inq.add((nx,ny))
    n=int(input())
    s=[input() for _ in range(n)]
    m=len(s[0])
    f=0
    for i in range(n):
        if f:
            break
        for j in range(m):
            if s[i][j]=='1':
                f=1
                start=bfs(i,j)
                break
    step=bfs2(start)
    print(step)
```

基本信息

#: 47584462
题目: 20741
提交人: 24n2400016635
内存: 4484kB
时间: 39ms
语言: Python3
提交时间: 2024-12-06 15:00:18

T28776: 国王游戏

greedy, <http://cs101.openjudge.cn/practice/28776>

思路:

(考后看的题解)我觉得这种对乘积排序非常不显然, 非常不好猜, 我即使去洗了澡吃了好吃的也没有自己灵光一现想出来。看题解的证明可以看懂, 自己想不出来。

我问了我高中同学, 他直接给我发了一道类似题目 (<https://www.luogu.com.cn/problem/P1842>), 并说很好猜, 因为注意到显然要把左手、右手数字大的放在最后, 再看到题目是除法, 所以要对乘积排序。??? 我只能想到用


```
lambda x:(x[0],x[1])
```

来分别排序，怎么显然地看出来对乘积排序的阿。类似题目是把乘法除法变成了加法减法。

代码：

```
n=int(input())
l,_=map(int,input().split())
s=[]
for _ in range(n):
    a,b=map(int,input().split())
    s.append((a*b,a,b))
s.sort(key=lambda x:x[0])
ans=l//s[0][2]
for i in range(1,n):
    l*=s[i-1][1]
    ans=max(ans,l//s[i][2])
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47617629提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n=int(input())
l,_=map(int,input().split())
s=[]
for _ in range(n):
    a,b=map(int,input().split())
    s.append((a*b,a,b))
s.sort(key=lambda x:x[0])
ans=l//s[0][2]
for i in range(1,n):
    l*=s[i-1][1]
    ans=max(ans,l//s[i][2])
print(ans)
```

基本信息

#: 47617629
题目: 28776
提交人: 24n2400016635
内存: 3624kB
时间: 20ms
语言: Python3
提交时间: 2024-12-07 23:54:34

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

2. 学习总结和收获

<mark>如果作业题目简单，有否额外练习题目，比如：OJ“计概 2024fall 每日选做”、CF、LeetCode、洛谷等网站题目。</mark>

最大的收获是考试一定要冷静和保持头脑清醒，刚开始太慌了第一题没做出来跳了（服了），然后第二题又超时，很慌。但是到最后 40min 的时候心静下来了，孤岛题代码居然可以一遍敲出来并 ac，最后才用非常复杂的思路去做第一题。这也说明心态会影响思考，在第一遍做题失败后很难想出简单的方法去做。

greedy 真不会，别人说的很好猜我真猜不出来，我只知道一般贪心题要排序，然后之后的处理要随题目而变，好难。dp 题有类似题目，难的是对设置的 dp 的含义要明确，才能写出准确的转移方程，还有初始化的考虑，也好难。双十一的 dfs 要设置好参数的意义，设置好退出条件，其余的代码部分就和八皇后的实现很相似了。

每日选做好难，做的有点慢，有好多新的知识，去力扣专门做了单调栈的题，最后十几天要冲刺冲刺 **dp** 的题目！（至少是我有可能做出来的，贪心太难了）