

月考 AC2 (悲)

1. (9min)

E22548: 机智的股民老张

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

思路：再从后往前更新另一个列表 b，b[i]表示第 i 天以后股票的峰值

代码：

```
a=list(map(int,input().split()))
n=len(a)
b=[0]*n
b[n-1]=a[n-1]
for j in range(n-2,-1,-1):
    b[j]=max(b[j+1],a[j])
num=0
for k in range(n):
    num=max(num,b[k]-a[k])
print(num)
```

运行：

状态: Accepted

源代码

```
a=list(map(int,input().split()))
n=len(a)
b=[0]*n
b[n-1]=a[n-1]
for j in range(n-2,-1,-1):
    b[j]=max(b[j+1],a[j])
num=0
for k in range(n):
    num=max(num,b[k]-a[k])
print(num)
```

基本信息

#: 47566526

题目: E22548

提交人: 24n2400011028

内存: 9568kB

时间: 55ms

语言: Python3

提交时间: 2024-12-05 15:17:16

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

2. (30min)

M28701: 炸鸡排

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

思路：感觉是困难的，必须注意到如果每个鸡排的时长都不大于

平均时长那么就可以煎完，并且锅必须同时煎 k 个鸡排其实和锅可以同时煎任意大于等于 k 个鸡排是等价的

代码：

```
n, k = map(int, input().split())
t = list(map(int, input().split()))
t=sorted(t)
s = sum(t)
while True:
    if t[-1] > s / k:
        s -= t.pop()
        k -= 1
    else:
        print('%.3f'%(s/k))
        break
```

运行：

#47667413提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n, k = map(int, input().split())
t = list(map(int, input().split()))
t=sorted(t)
s = sum(t)
while True:
    if t[-1] > s / k:
        s -= t.pop()
        k -= 1
    else:
        print('%.3f'%(s/k))
        break
```

基本信息

#: 47667413
题目: 28701
提交人: 24n2400011028
内存: 3620kB
时间: 29ms
语言: Python3
提交时间: 2024-12-10 18:30:41

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

3. (30min)

M20744: 土豪购物

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

思路：和第一题感觉有点类似，再去维护两个列表，记录从第 i 个物品开始向左/向右可以买到的最大价值

代码：

```

a=list(map(int,input().split(',')))
n=len(a)
left=[0]*n
right=[0]*n
if max(a)<=0:
    print(max(a))
else:
    answer=max(a)
    left[0]=max(a[0],0)
    for i in range(1,n):
        left[i]=max(left[i-1]+a[i],a[i],0)
    right[n-1]=a[n-1]
    for j in range(n-2,-1,-1):
        right[j]=max(right[j+1]+a[j],a[j],0)
    for k in range(1,n-1):
        if a[k]>=0:
            answer=max(answer,left[k-1]+a[k]+right[k+1])
        if a[k]<0:
            answer=max(answer,left[k-1]+right[k+1])
    print(answer)

```

运行:

状态: Accepted

源代码

```

a=list(map(int,input().split(',')))
n=len(a)
left=[0]*n
right=[0]*n
if max(a)<=0:
    print(max(a))
else:
    answer=max(a)
    left[0]=max(a[0],0)
    for i in range(1,n):
        left[i]=max(left[i-1]+a[i],a[i],0)
    right[n-1]=a[n-1]
    for j in range(n-2,-1,-1):
        right[j]=max(right[j+1]+a[j],a[j],0)
    for k in range(1,n-1):
        if a[k]>=0:
            answer=max(answer,left[k-1]+a[k]+right[k+1])
        if a[k]<0:
            answer=max(answer,left[k-1]+right[k+1])
    print(answer)

```

基本信息

#: 47567510
 题目: M20744
 提交人: 24n2400011028
 内存: 9432kB
 时间: 89ms
 语言: Python3
 提交时间: 2024-12-05 15:53:39

4. (30min)

T25561: 2022 决战双十一

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

思路: 没什么思维含量, 只是要考虑全面, dfs 即可。

代码:

```
#pylint:skip-file
def dfs(i, sumi):
    global cost, answer
    if i < n:
        for pri in price[i]:
            store, val = map(int, pri.split(':'))
            cost[store-1] += val
            dfs(i+1, sumi+val)
            cost[store-1] -= val
    else:
        save = 0
        for j in range(m):
            savej = 0
            for bon in bonus[j]:
                over, cut = map(int, bon.split('-'))
                if cost[j] >= over:
                    savej = max(savej, cut)
            save += savej
        answer = min(answer, sumi - save - (sumi // 300) * 50)
    return

n, m = map(int, input().split())
price = [input().split() for _ in range(n)]
bonus = [input().split() for _ in range(m)]
cost = [0] * m
answer = float('inf')
dfs(0, 0)
print(answer)
```

运行:

状态: Accepted

源代码

```
#pylint:skip-file
def dfs(i, sumi):
    global cost, answer
    if i < n:
        for pri in price[i]:
            store, val = map(int, pri.split(':'))
            cost[store-1] += val
            dfs(i+1, sumi+val)
            cost[store-1] -= val
    else:
        save = 0
        for j in range(m):
            savej = 0
            for bon in bonus[j]:
                over, cut = map(int, bon.split('-'))
                if cost[j] > over:
                    savej = max(savej, cut)
            save += savej
        answer = min(answer, sumi - save - (sumi//300)*50)
    return

n, m = map(int, input().split())
price = [input().split() for _ in range(n)]
bonus = [input().split() for _ in range(m)]
cost = [0]*m
answer = float('inf')
dfs(0, 0)
print(answer)
```

基本信息

#: 47673637
题目: 25561
提交人: 24n2400011028
内存: 3664kB
时间: 68ms
语言: Python3
提交时间: 2024-12-10 22:54:57

5. (1h+)

T20741: 两座孤岛最短距离

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

思路：考场上用两个广搜把两座岛都找出来，再直接枚举最小曼哈顿距离，结果超时了。回去后才看到可以用 Dijkstra 算法

代码：

```
import heapq
def dfs(x, y):
    visited = [[False]*m for _ in range(n)]
    q = []
    heapq.heappush(q, (0, x, y))
    dir = [[0, 1], [0, -1], [-1, 0], [1, 0]]
    while q:
        step, x, y = heapq.heappop(q)
        if visited[x][y]:
            continue
        visited[x][y] = True
        if mat[x][y] == '1' and step != 0:
            return step
        for dx, dy in dir:
            nx, ny = x+dx, y+dy
            if 0 <= nx <= n-1 and 0 <= ny <= m-1 and
```

```

visited[nx][ny]==False:
    heapq.heappush(q, (step+1-
int(mat[nx][ny]),nx,ny))
n=int(input())
mat=[]
for i in range(n):
    mat.append(input())
m=len(mat[0])
for c in range(n):
    for d in range(m):
        if mat[c][d]=='1':
            print(dfs(c,d))
            exit()

```

运行：

#47667096提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

import heapq
def dfs(x,y):
    visited=[[False]*m for _ in range(n)]
    q=[]
    heapq.heappush(q, (0,x,y))
    dir=[[0,1],[0,-1],[-1,0],[1,0]]
    while q:
        step,x,y=heapq.heappop(q)
        if visited[x][y]:
            continue
        visited[x][y]=True
        if mat[x][y]=='1' and step!=0:
            return step
        for dx,dy in dir:
            nx,ny=x+dx,y+dy
            if 0<=nx<=n-1 and 0<=ny<=m-1 and visited[nx][ny]==False:
                heapq.heappush(q, (step+1-int(mat[nx][ny]),nx,ny))
n=int(input())
mat=[]
for i in range(n):
    mat.append(input())
m=len(mat[0])
for c in range(n):
    for d in range(m):
        if mat[c][d]=='1':
            print(dfs(c,d))
            exit()

```

基本信息

#: 47667096
 题目: 20741
 提交人: 24n2400011028
 内存: 3700kB
 时间: 37ms
 语言: Python3
 提交时间: 2024-12-10 18:10:14

6. (40min)

T28776: 国王游戏

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

思路：也是不好想的，在题解的辅助下才想明白 greedy 策略。

按 $A_i \cdot B_i$ 排序。

代码：

```
n=int(input())
a,b=map(int,input().split())
number=[]
for _ in range(n):
    number.append(list(map(int,input().split())))
number=sorted(number,key=lambda x:(x[0]*x[1]))
result=0
for i in range(n):
    result=max(result,a//number[i][1])
    a*=number[i][0]
print(result)
```

运行:

#47668823提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

基本信息

源代码

```
n=int(input())
a,b=map(int,input().split())
number=[]
for _ in range(n):
    number.append(list(map(int,input().split())))
number=sorted(number,key=lambda x:(x[0]*x[1]))
result=0
for i in range(n):
    result=max(result,a//number[i][1])
    a*=number[i][0]
print(result)
```

#: 47668823

题目: 28776

提交人: 24n2400011028

内存: 3608kB

时间: 21ms

语言: Python3

提交时间: 2024-12-10 19:46:52

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

学习总结和收获:

还是深感 greedy 之难……有时候不知道是应该先试图去证明还是先把不知道正确性的代码写出来, 因为两者都会耗费一定时间。

马上机考了, 最后冲刺一下。