

Assignment #B: Dec Mock Exam 大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Compiled by <mark>同学的姓名、院系</mark>

说明:

1) 月考: AC6<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: 唯一一个能做的题, 先将 `min_prices` 初始化为无穷大, 把 `profit` 初始化为 `0`, 然后遍历每个 `li` 里的元素, 对 `min_prices` 与 `max_profit` 进行迭代, 最后输出即可 (貌似也可以用 `dp` 列表形式解决) (25min)

代码:

```
```python
li=list(map(int,input().split()))
min_prices=float('inf')
max_profit=0
```

```
for i in li:

 min_prices=min(min_prices,i)

 max_profit=max(max_profit,i-min_prices)

print(max_profit)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47601617提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
li=list(map(int,input().split()))
min_prices=float('inf')
max_profit=0
for i in li:
 min_prices=min(min_prices,i)
 max_profit=max(max_profit,i-min_prices)
print(max_profit)
```

基本信息

#: 47601617  
题目: 22548  
提交人: 24n2400011009  
内存: 9564kB  
时间: 47ms  
语言: Python3  
提交时间: 2024-12-07 10:40:30

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

### M28701: 炸鸡排

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

思路：对炸鸡排的时长进行排序，若时间需要最长的那一块比平均时间还长，移除，同时把锅中可容纳的鸡排数减去 1；

若时间需要最长的那块比平均时间还短，直接取平均时间即可，注意格式化输出（保留三位小数）

(2h)

代码：

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

代码运行截图 ==（至少包含有"Accepted"）==

状态: **Accepted**

源代码

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

基本信息

#: 47602134  
题目: 28701  
提交人: 24n2400011009  
内存: 3616kB  
时间: 21ms  
语言: Python3  
提交时间: 2024-12-07 11:00:13

### M20744: 土豪购物

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

思路: 创建了两个 **dp** 列表, 一个是用于表示从开头取到当前位置且不放回商品的`最大价值`, **dp2** 则是用来表示要取出一个商品后的`最大价值`


为什么两个递归的 `max` 里面都要有 `a[i]`? -> 因为防止前面的部分产生的负面影响, 这里有点像重新开始的意思

最后按题目要求输出 `max(dp2)` 即可 (**2h30min**)

代码:

```
```python
a = list(map(int, input().split(',')))
dp1 = [0] * len(a);
dp2 = [0] * len(a)
dp1[0] = a[0];
dp2[0] = a[0]
for i in range(1, len(a)):
    dp1[i] = max(dp1[i - 1] + a[i], a[i])
    dp2[i] = max(dp1[i - 1], dp2[i - 1] + a[i], a[i])
print(max(dp2))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



题目 排名 状态 提问

#47599891提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
a = list(map(int, input().split(',')))
dp1 = [0] * len(a)
dp2 = [0] * len(a)
dp1[0] = a[0]
dp2[0] = a[0]
for i in range(1, len(a)):
    dp1[i] = max(dp1[i - 1] + a[i], a[i])
    dp2[i] = max(dp1[i - 1], dp2[i - 1] + a[i], a[i])
print(max(dp2))
```

基本信息

#: 47599891

题目: 20744

提交人: 24n2400011009

内存: 9440kB

时间: 65ms

语言: Python3

提交时间: 2024-12-07 09:31:26

©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：先是初始化相关变量，然后再调用 `dfs` 函数进行深度搜索，搜索终止条件及计算优惠后的花费，再进行搜索过程中的状态拓展，最后调用 `dfs` 函数并输出结果(3h)

代码:

```
result = float("inf")

n, m = map(int, input().split())

store_prices = [input().split() for _ in range(n)]
coupons = [input().split() for _ in range(m)]
```

```
def dfs(store_prices, coupons, items=0, total_price=0,
each_store_price=[0] * m):

    global result

    if items == n:

        coupon_price = 0

        for i in range(m):

            store_p = 0

            for coupon in coupons[i]:

                a, b = map(int, coupon.split('-'))

                if each_store_price[i] >= a:

                    store_p = max(store_p, b)

            coupon_price += store_p
```

```
        result = min(result, total_price - (total_price
// 300) * 50 - coupon_price)
```



```
return
```

```
for i in store_prices[items]:  
    idx, p = map(int, i.split(':'))  
    each_store_price[idx - 1] += p  
    dfs(store_prices, coupons, items + 1, total_price  
+ p, each_store_price)  
    each_store_price[idx - 1] -= p
```

```
dfs(store_prices, coupons)  
print(result)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
result = float("inf")
n, m = map(int, input().split())
store_prices = [input().split() for _ in range(n)]
coupons = [input().split() for _ in range(m)]

def dfs(store_prices, coupons, items=0, total_price=0, each_store_price):
    global result
    if items == n:
        coupon_price = 0
        for i in range(m):
            store_p = 0
            for coupon in coupons[i]:
                a, b = map(int, coupon.split('-'))
                if each_store_price[i] >= a:
                    store_p = max(store_p, b)

            coupon_price += store_p

        result = min(result, total_price - (total_price // 300) * 50 - coupon_price)
        return

    for i in store_prices[items]:
        idx, p = map(int, i.split(':'))
        each_store_price[idx - 1] += p
        dfs(store_prices, coupons, items + 1, total_price + p, each_store_price)
        each_store_price[idx - 1] -= p
```

基本信息

#: 47601224
题目: 25561
提交人: 24n2400011009
内存: 3668kB
时间: 65ms
语言: Python3
提交时间: 2024-12-07 10:27:52

T20741: 两座孤岛最短距离

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

思路: 主要分成 dfs, bfs, main 三个函数, dfs 用于找到孤岛并将孤岛添加到 queue 中 (遍历四个方向, 标记 grid[x][y])

bfs 主要用来找到最短距离 (while queue->搜索 dx, dy 并判断是否是水域, 若不是直接 return distance, 若是则将

其继续增添至 queue 中继续搜索 (注意 distance+=1 的缩进, 与 bfs 的原理相关, 即已经把这一层的所有节点搜索完了)

main 函数主要用来进行数据输入 (n,grid,queue)，以及当 grid[i][j] 是陆地时开始进行 dfs 搜索并返回 bfs 搜索的值(3h)

代码:

```
```python
```

```
from collections import deque
```

```
def dfs(x, y, grid, n, queue, directions):
 grid[x][y] = 2
 queue.append((x, y))
 for dx, dy in directions:
 nx, ny = x + dx, y + dy
 if 0 <= nx < n and 0 <= ny < n and grid[nx][ny]
== 1:
 dfs(nx, ny, grid, n, queue, directions)
```

```
def bfs(grid, n, queue, directions):
 distance = 0
 while queue:
```

```

 for _ in range(len(queue)):
 x, y = queue.popleft()
 for dx, dy in directions:
 nx, ny = x + dx, y + dy
 if 0 <= nx < n and 0 <= ny < n:
 if grid[nx][ny] == 1:
 return distance
 elif grid[nx][ny] == 0:
 grid[nx][ny] = 2
 queue.append((nx, ny))

 distance += 1
 return distance

```

```

def main():
 n = int(input())
 grid = [list(map(int, input())) for _ in range(n)]
 directions = [(1, 0), (-1, 0), (0, 1), (0, -1)]
 queue = deque()
 for i in range(n):
 for j in range(n):
 if grid[i][j] == 1:
 dfs(i, j, grid, n, queue, directions)

```

```
return bfs(grid, n, queue, directions)
```

```
if __name__ == "__main__":
 print(main())
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47600326提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
from collections import deque

def dfs(x, y, grid, n, queue, directions):
 grid[x][y] = 2
 queue.append((x, y))
 for dx, dy in directions:
 nx, ny = x + dx, y + dy
 if 0 <= nx < n and 0 <= ny < n and grid[nx][ny] == 1:
 dfs(nx, ny, grid, n, queue, directions)

def bfs(grid, n, queue, directions):
 distance = 0
 while queue:
 for _ in range(len(queue)):
 x, y = queue.popleft()
 for dx, dy in directions:
 nx, ny = x + dx, y + dy
 if 0 <= nx < n and 0 <= ny < n:
 if grid[nx][ny] == 1:
 return distance
 elif grid[nx][ny] == 0:
 grid[nx][ny] = 2
 queue.append((nx, ny))
 distance += 1
 return distance
```

基本信息

#: 47600326  
题目: 20741  
提交人: 24n2400011009  
内存: 4300kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-12-07 09:53:47

### T28776: 国王游戏

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

"""

思路：这道题最关键的是推导出当一个大臣左右手乘积>后面的大臣的左右手乘积时,交换这两个大臣，可以得到最大答案的最小值（题解太妙了）

然后问题便可迎刃而解。

对于 `solution` 函数，先利用匿名函数对 `lst` 列表进行排序，然后不断更新 `ans` 的值与 `a` 的值（前面的人的左手上的乘积发生改变），最后 `return ans`

对于主函数即为输入部分

（真的非常非常难想！！！）（2h）

代码：

```
from typing import List

def Solution(n:int, a:int, b:int, lst:List[List]) -> int:
 lst.sort(key=lambda x: (x[0] * x[1]))
 ans = 0
 for i in range(n):
 ans = max(ans, a // lst[i][1])
 a *= lst[i][0]
 return ans
```

```
if __name__ == "__main__":
 n = int(input())
 a, b = map(int, input().split())
 lst = []
 for i in range(n):
 lst.append([int(_) for _ in input().split()])
 print(Solution(n, a, b, lst))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#47600777提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from typing import List
def Solution(n:int, a:int, b:int, lst:List[List]) -> int:
 lst.sort(key=lambda x: (x[0] * x[1]))
 ans = 0
 for i in range(n):
 ans = max(ans, a // lst[i][1])
 a *= lst[i][0]
 return ans
if __name__ == "__main__":
 n = int(input())
 a, b = map(int, input().split())
 lst = []
 for i in range(n):
 lst.append([int(_) for _ in input().split()])
 print(Solution(n, a, b, lst))
```

基本信息

#: 47600777  
题目: 28776  
提交人: 24n2400011009  
内存: 4132kB  
时间: 24ms  
语言: Python3  
提交时间: 2024-12-07 10:11:41

©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>

月考的时候也就会做第一题，后面几个题实在太难了，就算把题解看懂了应该到了期末还是做不出来，这段时间开始复习之前做过的一些作业，

然后做了一些 leetcode 上的题，一定要把 easy 和 medium 上的题做对啊啊啊啊

感觉现在是 dp 和 greedy 想不出来，dfs 和 bfs 套模板很容易遗漏一些东西，得 debug 很久，所以等效于好像什么都不会 qwq