Assignment #B: Dec Mock Exam大雪前一天

Updated 1649 GMT+8 Dec 5, 2024

2024 fall, Complied by 李振硕、信息管理系

1. 题目

E22548: 机智的股民老张

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

思路：

代码：

状态: Accepted

源代码

```
a=list(map(int,input().split()))
a2=max(a)
max2=0
for i in range(len(a)):
    if a[i]<=a2:
        a2=a[i]
        if max(a[i+1:])-a[i]>max2:
            max2=max(a[i+1:])-a[i]
print(max2)
```

基本信息

#: 47566694
题目: E22548
提交人: 24n2300093007
内存: 9576kB
时间: 48ms
语言: Python3
提交时间: 2024-12-05 15:22:25

M28701: 炸鸡排

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

思路：

代码：

#47673585提交状态

查看　提交　统计　提问

状态: Accepted

源代码

```
def calculate_max_frying_time(n, k, frying_times):
    # 按炸熟时间升序排序
    frying_times.sort()
    total_time = sum(frying_times)

    while True:
        # 如果当前炸锅中最大的鸡排时间超过平均分配时间
        if frying_times[-1] > total_time / k:
            total_time -= frying_times.pop()  # 移除当前最长时间
            k -= 1  # 减少炸锅容量
        else:
            # 输出最大持续时间
            return f"{total_time / k:.3f}"


n, k = map(int, input().split())
frying_times = list(map(int, input().split()))

print(calculate_max_frying_time(n, k, frying_times))
```

基本信息

#: 47673585
题目: 28701
提交人: 24n2300093007
内存: 3612kB
时间: 22ms
语言: Python3
提交时间: 2024-12-10 22:52:42

English　帮助　关于

M20744: 土豪购物

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

思路：

代码：

T25561: 2022决战双十一

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

思路：

代码：

状态：Accepted

基本信息

#：47672299
题目：25561
提交人：24n2300093007
内存：3704kB
时间：38ms
语言：Python3
提交时间：2024-12-10 21:50:49

源代码

```
from itertools import product
from math import inf

def min_price(n, m, prices, coupons):
    store_prices = [[] for _ in range(n)]
    for i in range(n):
        for price_info in prices[i].split():
            store, price = map(int, price_info.split(":"))
            store_prices[i].append((store - 1, price))
    store_coupons = []
    for coupon_list in coupons:
        store_coupons.append([])
        for coupon in coupon_list.split():
            q, x = map(int, coupon.split("-"))
            store_coupons[-1].append((q, x))

    min_total_cost = inf

    for combination in product(*[range(len(store_prices[i])) for i in r
        store_totals = [0] * m
        for i, choice in enumerate(combination):
            store, price = store_prices[i][choice]
            store_totals[store] += price

        discounted_totals = store_totals[:]
        for store in range(m):
            max_discount = 0
            for q, x in store_coupons[store]:
                if store_totals[store] >= q:
                    max_discount = max(max_discount, x)
            discounted_totals[store] -= max_discount

        total_cost = sum(discounted_totals)
        total_raw_price = sum(store_totals)
        total_cost -= (total_raw_price // 300) * 50

        min_total_cost = min(min_total_cost, total_cost)

    return min_total_cost

n, m = map(int, input().split())
prices = [input() for _ in range(n)]
coupons = [input() for _ in range(m)]

print(min_price(n, m, prices, coupons))
```

T20741: 两座孤岛最短距离

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

思路：

代码：

状态: Accepted

基本信息

#: 47673133
题目: 20741
提交人: 24n2300093007
内存: 3944kB
时间: 31ms
语言: Python3
提交时间: 2024-12-10 22:31:29

源代码

```
from collections import deque

def shortest_bridge(n, grid):
    def get_islands():
        def dfs(x, y, marker):
            stack = [(x, y)]
            grid[x][y] = marker
            island_points.append((x, y))
            while stack:
                cx, cy = stack.pop()
                for dx, dy in directions:
                    nx, ny = cx + dx, cy + dy
                    if 0 <= nx < n and 0 <= ny < n and grid[nx][ny] ==
                        grid[nx][ny] = marker
                        island_points.append((nx, ny))
                        stack.append((nx, ny))

        islands = []
        marker = 2
        for i in range(n):
            for j in range(n):
                if grid[i][j] == 1:
                    island_points = []
                    dfs(i, j, marker)
                    islands.append(island_points)
                    marker += 1
        return islands

    def bfs_from_island(island):
        queue = deque()
        visited = [[False] * n for _ in range(n)]
        for x, y in island:
            queue.append((x, y, 0))
            visited[x][y] = True

        while queue:
            x, y, dist = queue.popleft()
            for dx, dy in directions:
                nx, ny = x + dx, y + dy
                if 0 <= nx < n and 0 <= ny < n:
                    if not visited[nx][ny]:
                        if grid[nx][ny] == 3:
                            return dist
                        elif grid[nx][ny] == 0:
                            visited[nx][ny] = True
                            queue.append((nx, ny, dist + 1))
        return float('inf')


    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

    islands = get_islands()

    return bfs_from_island(islands[0])

n = int(input())
grid = [list(map(int, input().strip())) for _ in range(n)]

print(shortest_bridge(n, grid))
```

T28776: 国王游戏

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

思路：

代码：

状态: Accepted

源代码

```
def minimize_max_gold(n, king, ministers):

    a0, b0 = king
    ministers.sort(key=lambda x: x[0] * x[1])
    result = 0

    for a, b in ministers:
        # 计算当前大臣的金币数
        result = max(result, a0 // b)
        # 更新前缀乘积
        a0 *= a

    return result

n = int(input().strip())
king = tuple(map(int, input().strip().split()))
ministers = [tuple(map(int, input().strip().split())) for _ in range(n)]

print(minimize_max_gold(n, king, ministers))
```

基本信息

#:	47673504
题目:	28776
提交人:	24n2300093007
内存:	3644kB
时间:	22ms
语言:	Python3
提交时间:	2024-12-10 22:48:52

2. 学习总结和收获

这次月考真的很难，，，只对了一道题，第二题提交了十多次，但一直是runtime error，应该机考之前复习好，再考。。