Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 刘家亦、物院

说明：

1）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。

2）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。

3）如果不能在截止前提交作业，请写明原因。

1. 题目

1115. 取石子游戏

dfs, https://www.acwing.com/problem/content/description/1117/

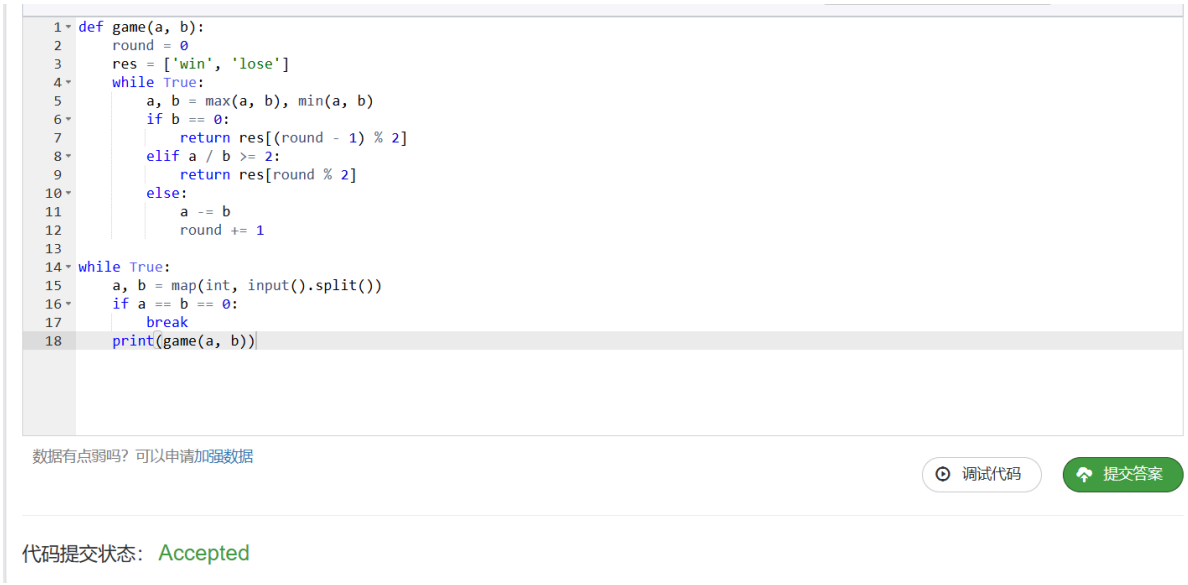
思路：按照提示做即可。这是一道非常简单的贪心，贪心策略就算没有提示也很好想到：如果$\frac{a}{b} >= 2$，那么如果先手直接把能拿的拿完了会输，那么先手就留一个让后手拿，此时后手没有选择，只能拿，此时将必输的结局转化给了后手。

代码：

```
def game(a, b):
    round = 0
    res = ['win', 'lose']
    while True:
        a, b = max(a, b), min(a, b)
        if b == 0:
            return res[(round - 1) % 2]
        elif a / b >= 2:
            return res[round % 2]
        else:
            a -= b
            round += 1

while True:
    a, b = map(int, input().split())
    if a == b == 0:
        break
    print(game(a, b))
```

代码运行截图（至少包含有"Accepted"）



```python
def game(a, b):
    round = 0
    res = ['win', 'lose']
    while True:
        a, b = max(a, b), min(a, b)
        if b == 0:
            return res[(round - 1) % 2]
        elif a / b >= 2:
            return res[round % 2]
        else:
            a -= b
            round += 1

while True:
    a, b = map(int, input().split())
    if a == b == 0:
        break
    print(game(a, b))
```


25570: 洋葱

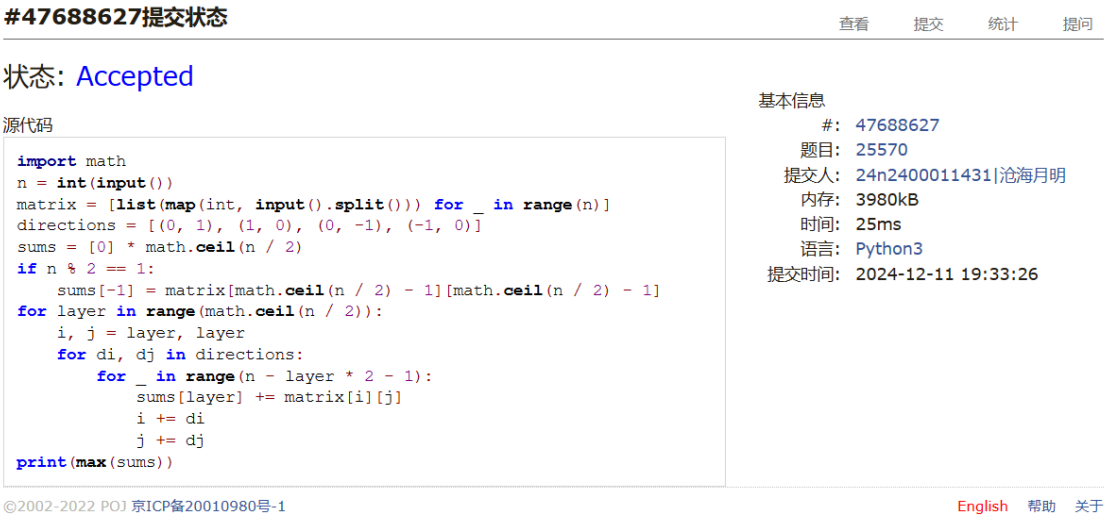
Matrices, http://cs101.openjudge.cn/practice/25570

思路：和螺旋矩阵类似，只要见过一次就不新鲜了

代码：

```python
import math
n = int(input())
matrix = [list(map(int, input().split())) for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
sums = [0] * math.ceil(n / 2)
if n % 2 == 1:
    sums[-1] = matrix[math.ceil(n / 2) - 1][math.ceil(n / 2) - 1]
for layer in range(math.ceil(n / 2)):
    i, j = layer, layer
    for di, dj in directions:
        for _ in range(n - layer * 2 - 1):
            sums[layer] += matrix[i][j]
            i += di
            j += dj
print(max(sums))
```

代码运行截图 ==（至少包含有"Accepted"）==



#47688627提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import math
n = int(input())
matrix = [list(map(int, input().split())) for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
sums = [0] * math.ceil(n / 2)
if n % 2 == 1:
    sums[-1] = matrix[math.ceil(n / 2) - 1][math.ceil(n / 2) - 1]
for layer in range(math.ceil(n / 2)):
    i, j = layer, layer
    for di, dj in directions:
        for _ in range(n - layer * 2 - 1):
            sums[layer] += matrix[i][j]
            i += di
            j += dj
print(max(sums))
```

基本信息

#: 47688627
题目: 25570
提交人: 24n2400011431|沧海月明
内存: 3980kB
时间: 25ms
语言: Python3
提交时间: 2024-12-11 19:33:26

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

1526C1. Potions(Easy Version)

greedy, dp, data structures, brute force, *1500, https://codeforces.com/problemset/problem/1526/C1

思路：非常朴素的贪心，朴素到我感觉这个不像是对的，结果验证了几组数据发现好像是对的。不过没想到答案最小堆的优美做法qwq

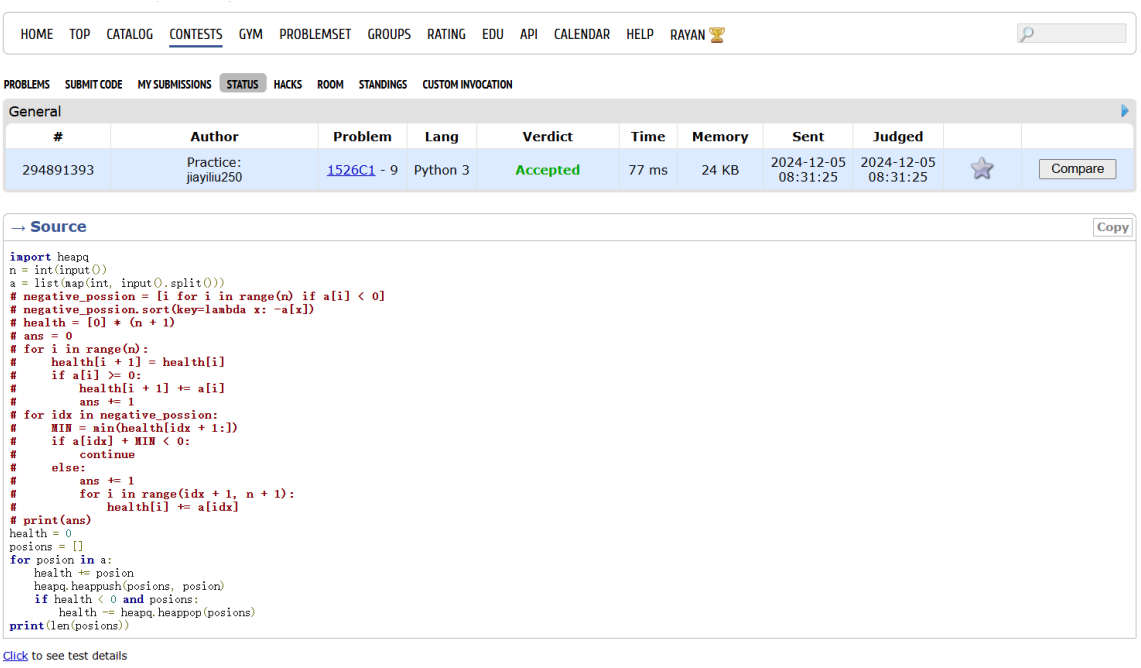
代码：

```
n = int(input())
a = list(map(int, input().split()))
negative_possion = [i for i in range(n) if a[i] < 0]
negative_possion.sort(key=lambda x: -a[x])
health = [0] * (n + 1)
ans = 0
for i in range(n):
    health[i + 1] = health[i]
    if a[i] >= 0:
        health[i + 1] += a[i]
        ans += 1
for idx in negative_possion:
    MIN = min(health[idx + 1:])
    if a[idx] + MIN < 0:
        continue
    else:
        ans += 1
        for i in range(idx + 1, n + 1):
            health[i] += a[idx]
print(ans)

# 最小堆
import heapq
n = int(input())
a = list(map(int, input().split()))
```

```
health = 0
posions = []
for posion in a:
    health += posion
    heapq.heappush(posions, posion)
    if health < 0 and posions:
        health -= heapq.heappop(posions)
print(len(posions))
```

代码运行截图 （至少包含有"Accepted"）



22067: 快速堆猪

辅助栈，http://cs101.openjudge.cn/practice/22067/

思路：就照着题目的描述硬打就行了，不过代码写的好丑orz

代码：

```
import heapq
stack = []; p = []
while True:
    try:
        s = input()
    except EOFError:
        break
    if s == 'min':
        if stack:
            print(p[0])
    elif s == 'pop':
        if stack:
            poped = stack.pop()
```

```
            if poped == p[0]:
                heapq.heappop(p)
    else:
        n = int(s.split()[1])
        stack.append(n)
        heapq.heappush(p, n)
```

代码运行截图 （至少包含有"Accepted"）

20106: 走山路

Dijkstra, http://cs101.openjudge.cn/practice/20106/

思路：很标准的Dijkstra算法

代码：

```
import heapq
def dijstra(x1, y1, x2, y2):
    q = [(0, x1, y1)]
    distances = [[float('inf')] * n for _ in range(m)]
    distances[x1][y1] = 0
    while q:
        distance, x, y = heapq.heappop(q)
        if distances[x][y] < distance:
            continue
        for dx, dy in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
            if 0 <= x + dx < m and 0 <= y + dy < n and matrix[x + dx][y + dy] !=
'#':
                new_distance = distance + abs(matrix[x + dx][y + dy] - matrix[x]
[y])
                if new_distance < distances[x + dx][y + dy]:
```

```
                    distances[x + dx][y + dy] = new_distance
                    heapq.heappush(q, (new_distance, x + dx, y + dy))
    return distances[x2][y2]

m, n, p = map(int, input().split())
matrix = [list(map(lambda x: int(x) if x.isdigit() else x, input().split())) for
_ in range(m)]
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    if matrix[x1][y1] == '#' or matrix[x2][y2] == '#':
        print('NO')
        continue

    distance = dijstra(x1, y1, x2, y2)
    if distance == float('inf'):
        print('NO')
    else:
        print(distance)
```

代码运行截图 （至少包含有"Accepted"）

#47511906提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
def dijstra(x1, y1, x2, y2):
    q = [(0, x1, y1)]
    distances = [[float('inf')] * n for _ in range(m)]
    distances[x1][y1] = 0
    while q:
        distance, x, y = heapq.heappop(q)
        if distances[x][y] < distance:
            continue
        for dx, dy in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
            if 0 <= x + dx < m and 0 <= y + dy < n and matrix[x + dx][y
                new_distance = distance + abs(matrix[x + dx][y + dy] - r
                if new_distance < distances[x + dx][y + dy]:
                    distances[x + dx][y + dy] = new_distance
                    heapq.heappush(q, (new_distance, x + dx, y + dy))
    return distances[x2][y2]

m, n, p = map(int, input().split())
matrix = [list(map(lambda x: int(x) if x.isdigit() else x, input().spli
for _ in range(p):
    x1, y1, x2, y2 = map(int, input().split())
    if matrix[x1][y1] == '#' or matrix[x2][y2] == '#':
        print('NO')
        continue

    distance = dijstra(x1, y1, x2, y2)
    if distance == float('inf'):
        print('NO')
    else:
        print(distance)
```

基本信息

#: 47511906
题目: 20106
提交人: 24n2400011431|沧海月明
内存: 3740kB
时间: 346ms
语言: Python3
提交时间: 2024-12-02 11:10:51

04129: 变换的迷宫

bfs, http://cs101.openjudge.cn/practice/04129/

思路：有趣的bfs，难点在于visited的更新不能是即时的。

代码：

```
from collections import deque
def bfs(r_begin, c_begin, matrix, k):
    q = deque([(r_begin, c_begin, 0)])
    visited = [[[False] * len(matrix[0]) for i in range(len(matrix))] for j in
range(k)]
    while q:
        r, c, depth = q.pop()
        if matrix[r][c] == 'E':
            return depth
        for dr, dc in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
            nr, nc = r + dr, c + dc
            if 0 <= nr < len(matrix) and 0 <= nc < len(matrix[0])\
                and (matrix[nr][nc] != '#' or (depth + 1) % k == 0) and not
visited[(depth + 1) % k][nr][nc]:
                    q.appendleft((nr, nc, depth + 1))
                    visited[(depth + 1) % k][nr][nc] = True
    return 'Oop!'

t = int(input())
for _ in range(t):
    r, c, k = map(int, input().split())
    matrix = [list(input()) for _ in range(r)]
    for i in range(r):
        for j in range(c):
            if matrix[i][j] == 'S':
                print(bfs(i, j, matrix, k))
```

代码运行截图 （至少包含有"Accepted"）

#47533983提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
def bfs(r_begin, c_begin, matrix, k):
    q = deque([(r_begin, c_begin, 0)])
    visited = [[[False] * len(matrix[0]) for i in range(len(matrix))] f
    while q:
        r, c, depth = q.pop()
        if matrix[r][c] == 'E':
            return depth
        for dr, dc in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
            nr, nc = r + dr, c + dc
            if 0 <= nr < len(matrix) and 0 <= nc < len(matrix[0])\
                and (matrix[nr][nc] != '#' or (depth + 1) % k == 0) and
                    q.appendleft((nr, nc, depth + 1))
                    visited[(depth + 1) % k][nr][nc] = True
    return 'Oop!'

t = int(input())
for _ in range(t):
    r, c, k = map(int, input().split())
    matrix = [list(input()) for _ in range(r)]
    for i in range(r):
        for j in range(c):
            if matrix[i][j] == 'S':
                print(bfs(i, j, matrix, k))
```

基本信息

#: 47533983
题目: 04129
提交人: 24n2400011431|沧海月明
内存: 4600kB
时间: 115ms
语言: Python3
提交时间: 2024-12-03 15:49:29

English 帮助 关于

2. 学习总结和收获

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。

最后两周了，冲冲冲！！！真得满分吧！
另：老师的作业似乎是用16进制标记的