Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 熊程宇 物理学院

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:

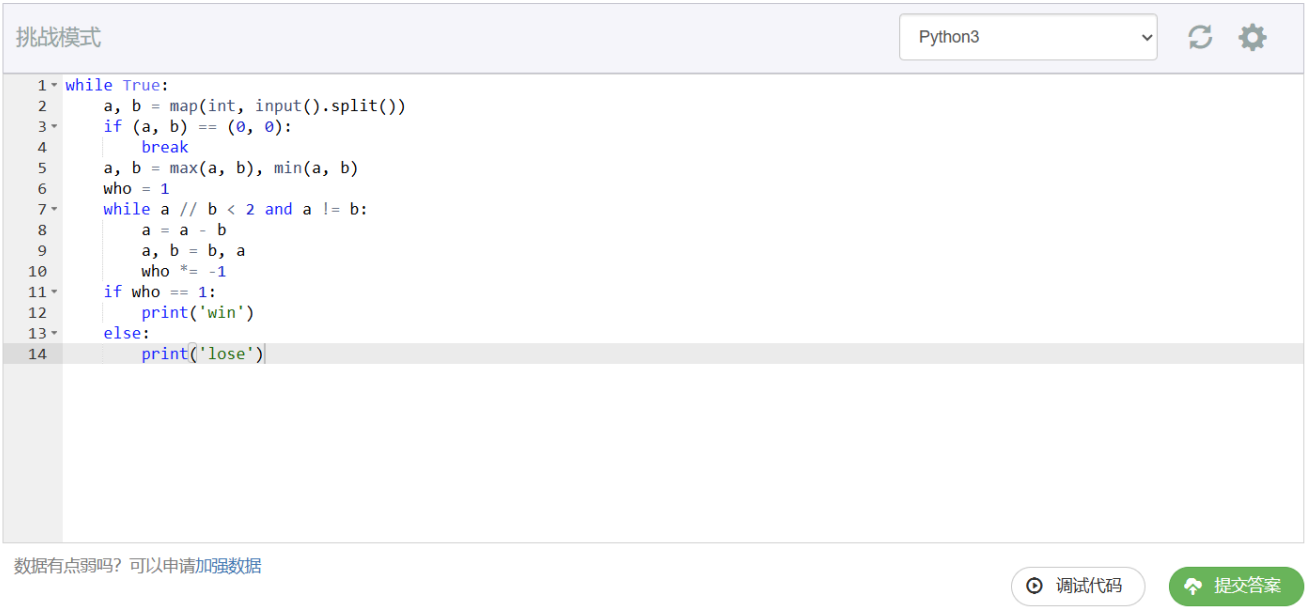
被dfs哄骗…

题目的提示极大地简化了这个问题，但是要注意讨论a == b这种特殊情况（需要循环的就是1<[a/b]<2)

代码:

```
while True:
    a, b = map(int, input().split())
    if (a, b) == (0, 0):
        break
    a, b = max(a, b), min(a, b)
    who = 1
    while a // b < 2 and a != b:
        a = a - b
        a, b = b, a
        who *= -1
    if who == 1:
        print('win')
    else:
        print('lose')
```

代码运行截图 （至少包含有"Accepted"）



代码提交状态： Accepted

25570: 洋葱

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

思路：

去外层，最后额外奇偶讨论即可

代码：

```
n = int(input())
mat = []
for _ in range(n):
    row = list(map(int, input().split()))
    mat.append(row)
max_layer = 0
sum_layer = 0
while len(mat) > 1:
    sum_layer = sum(mat[0]) + sum(mat[-1])
    mat.pop(0)
    mat.pop(-1)
    mat_new = []
    for row in mat:
        sum_layer += row[0] + row[-1]
        row.pop(0)
        row.pop(-1)
        mat_new.append(row)
    max_layer = max(max_layer, sum_layer)
    mat = mat_new
if len(mat) == 1:
    max_layer = max(max_layer, mat[0][0])
print(max_layer)
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n = int(input())
mat = []
for _ in range(n):
    row = list(map(int, input().split()))
    mat.append(row)
max_layer = 0
sum_layer = 0
while len(mat) > 1:
    sum_layer = sum(mat[0]) + sum(mat[-1])
    mat.pop(0)
    mat.pop(-1)
    mat_new = []
    for row in mat:
        sum_layer += row[0] + row[-1]
        row.pop(0)
        row.pop(-1)
        mat_new.append(row)
    max_layer = max(max_layer, sum_layer)
    mat = mat_new
if len(mat) == 1:
    max_layer = max(max_layer, mat[0][0])
print(max_layer)
```

基本信息

#:	47706455
题目:	25570
提交人:	24n2400011504
内存:	3976kB
时间:	22ms
语言:	Python3
提交时间:	2024-12-12 19:21:26

1526C1. Potions(Easy Version)

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

思路:

想了很久动态规划之后实现不出来，遂抄答案...

思路：对喝药水瓶数做动态规划，每喝一瓶新的药水，就更新各个喝药水瓶数的剩余健康值

代码:

```
n = int(input())
a = list(map(int ,input().split()))
dp = [-float('inf')] * (1 + n)
dp[0] = 0
for i in range(1 + n):
    for j in range(i, 0, -1):
        temp=max(dp[j], dp[j - 1] + a[i - 1])
        if temp >= 0:
            dp[j] = temp

for i in range(n, -1, -1):
    if dp[i] >= 0:
        print(i)
        break
```

代码运行截图 （至少包含有"Accepted"）

→ Last submissions

Submission	Time	Verdict
296177181	Dec/12/2024 18:03	Accepted
296177039	Dec/12/2024 18:02	Runtime error on test 1
296157457	Dec/12/2024 15:40	Wrong answer on test 2

22067: 快速堆猪

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

思路：

需要注意的就是处理堆顶就是最小值的情况

注意这种时候，正好就是记录过的最小值的最新一个，也就是最后一个，使用栈来弹出便合理了

代码：

```
pig = []
min_stack = []

while True:
    try:
        command = input().split()
        if command[0] == 'push':
            weight = int(command[1])
            pig.append(weight)
            if not min_stack or weight <= min_stack[-1]:
                min_stack.append(weight)
        elif command[0] == 'pop':
            if pig:
                removed = pig.pop()
                if min_stack and removed == min_stack[-1]:
                    min_stack.pop()
        elif command[0] == 'min':
            if min_stack:
                print(min_stack[-1])
    except EOFError:
        break
```

代码运行截图 （至少包含有"Accepted"）

#	结果	时间
4	Accepted	2024-12-12
3	Accepted	2024-12-12
2	Wrong Answer	2024-12-12
1	Runtime Error	2024-12-12

20106: 走山路

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

思路:

几乎就是把deque换成heapq

代码:

```
import heapq

def min_effort(m, n, terrain, test_cases):
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    def is_valid(x, y):
        return 0 <= x < m and 0 <= y < n and terrain[x][y] != "#"
    def dijkstra(start, end):
        if terrain[start[0]][start[1]] == "#" or terrain[end[0]][end[1]] == "#":
            return "NO"
        heap = [(0, start[0], start[1])]
        visited = set()
        while heap:
            cost, x, y = heapq.heappop(heap)
            if (x, y) in visited:
                continue
            visited.add((x, y))
            if (x, y) == end:
                return cost
            for dx, dy in directions:
                nx, ny = x + dx, y + dy
                if is_valid(nx, ny) and (nx, ny) not in visited:
                    new_cost = cost + abs(int(terrain[nx][ny]) - int(terrain[x][y]))
                    heapq.heappush(heap, (new_cost, nx, ny))
        return "NO"
    results = []
    for start, end in test_cases:
        results.append(dijkstra(start, end))
    return results
m, n, p = map(int, input().split())
terrain = [input().split() for _ in range(m)]
test_cases = [tuple(map(int, input().split())) for _ in range(p)]
test_cases = [((x1, y1), (x2, y2)) for x1, y1, x2, y2 in test_cases]
results = min_effort(m, n, terrain, test_cases)
for res in results:
    print(res)
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
import heapq

def min_effort(m, n, terrain, test_cases):
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

    def is_valid(x, y):
        return 0 <= x < m and 0 <= y < n and terrain[x][y] != "#"
```

基本信息

#: 47712942
题目: 20106
提交人: 24n2400011504
内存: 3896kB
时间: 220ms
语言: Python3
提交时间: 2024-12-13 10:46:30

04129: 变换的迷宫

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

思路:

首先必须要想到开三维visited进行标记，如果这个迷宫的变换逻辑是一般的，三维是必然的

而对于这种特殊变换，抄答案得知可以把visited的时间长度简化为k长度

在某个点遍历到E时就返回，没有必要把这个点存储，这就是莫名增加代码工作量.

代码:

```
from collections import deque
dir = [(0, 1), (0, -1), (1, 0), (-1, 0)]
def min_length(r, c, k, maze, start, end):
    visited = set()
    q = deque()
    q.append((0, start[0], start[1]))
    visited.add((0, start[0], start[1]))
    while q:
        time, x, y = q.popleft()
        for dx, dy in dir:
            nx, ny = x + dx, y + dy
            next_time = (time + 1) % k
            if 0 <= nx < r and 0 <= ny < c and (next_time, nx, ny) not in visited:
                c1 = maze[nx][ny]
                if c1 == 'E':
                    return time + 1
                elif c1 != '#' or (c1 == '#' and next_time == 0):
                    q.append((time + 1, nx, ny))
                    visited.add((next_time, nx, ny))
    return -1
t = int(input())
for _ in range(t):
    r, c, k = map(int, input().split())
    maze = [list(input().strip()) for _ in range(r)]
    start = None
    end = None
```

```
    for i in range(r):
        for j in range(c):
            if maze[i][j] == 'S':
                start = (i, j)
            elif maze[i][j] == 'E':
                end = (i, j)
    a = min_length(r, c, k, maze, start, end)
    print(a if a != -1 else 'Oop!')
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
from collections import deque
dir = [(0, 1), (0, -1), (1, 0), (-1, 0)]
def min_length(r, c, k, maze, start, end):
    visited = set()
    q = deque()
    q.append((0, start[0], start[1]))
    visited.add((0, start[0], start[1]))
    while q:
        time, x, y = q.popleft()
        for dx, dy in dir:
            nx, ny = x + dx, y + dy
            next_time = (time + 1) % k
            if 0 <= nx < r and 0 <= ny < c and (next_time, nx, ny) not 
                c1 = maze[nx][ny]
                if c1 == 'E':
                    return time + 1
                elif c1 != '#' or (c1 == '#' and next_time == 0):
                    q.append((time + 1, nx, ny))
                    visited.add((next_time, nx, ny))
    return -1
```

基本信息

#:	47712867
题目:	04129
提交人:	24n2400011504
内存:	5088kB
时间:	111ms
语言:	Python3
提交时间:	2024-12-13 10:41:23

2. 学习总结和收获

接下来得把作业重新复习一遍，再着重复习dp与bfs的变形.