Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 徐贤天，工学院

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：

如果a/b>=2或a=b，那么先手必赢；否则先手只有一种选择，接下来的情况类似。

代码：

```
def can_win(a, b, step):
    if a == b:
        return True
    elif a // b >= 2:
        return True
    else:
        new_a, new_b = b, a-b
        if new_a // new_b >= 2:
            if step % 2 == 0:
                return False
            else:
                return True
        else:
            return can_win(new_a, new_b, step+1)

while True:
    a, b = map(int, input().split())
    if a < b:
        a, b = b, a
    if a == b == 0:
        break
    print('win' if can_win(a, b, 0) else '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：

代码：

```
from math import ceil
n = int(input())
matrix = [list(map(int, input().split())) for _ in range(n)]
visited = [[False] * n for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
ans = []
for x in range(0, ceil(n/2)):
    ret = 0
    idx = 0
    i, j = x, x
    while 0 <= i < n and 0 <= j < n and not visited[i][j]:
        visited[i][j] = True
        ret += matrix[i][j]
        di, dj = directions[idx]
        ni, nj = i + di, j + dj
        if not (0 <= ni < n and 0 <= nj < n and not visited[ni][nj]):
            idx += 1
            if idx == 4:
                break
            else:
                di, dj = directions[idx]
                i, j = i + di, j + dj
        else:
```

```
            i, j = ni, nj
    ans.append(ret)
print(max(ans))
```

代码运行截图 ==（至少包含有"Accepted"）==

#47676249提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from math import ceil
n = int(input())
matrix = [list(map(int, input().split())) for _ in range(n)]
visited = [[False] * n for _ in range(n)]
directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
ans = []
for x in range(0, ceil(n/2)):
    ret = 0
    idx = 0
    i, j = x, x
    while 0 <= i < n and 0 <= j < n and not visited[i][j]:
        visited[i][j] = True
        ret += matrix[i][j]
        di, dj = directions[idx]
        ni, nj = i + di, j + dj
        if not (0 <= ni < n and 0 <= nj < n and not visited[ni][nj]):
            idx += 1
            if idx == 4:
                break
            else:
                di, dj = directions[idx]
                i, j = i + di, j + dj
        else:
            i, j = ni, nj
    ans.append(ret)
print(max(ans))
```

基本信息

#: 47676249
题目: 25570
提交人: 24n2400011033
内存: 4056kB
时间: 29ms
语言: Python3
提交时间: 2024-12-11 09:39:21

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:

把能吃的都吃，吃太多了再吐出来最小的（收益最低）

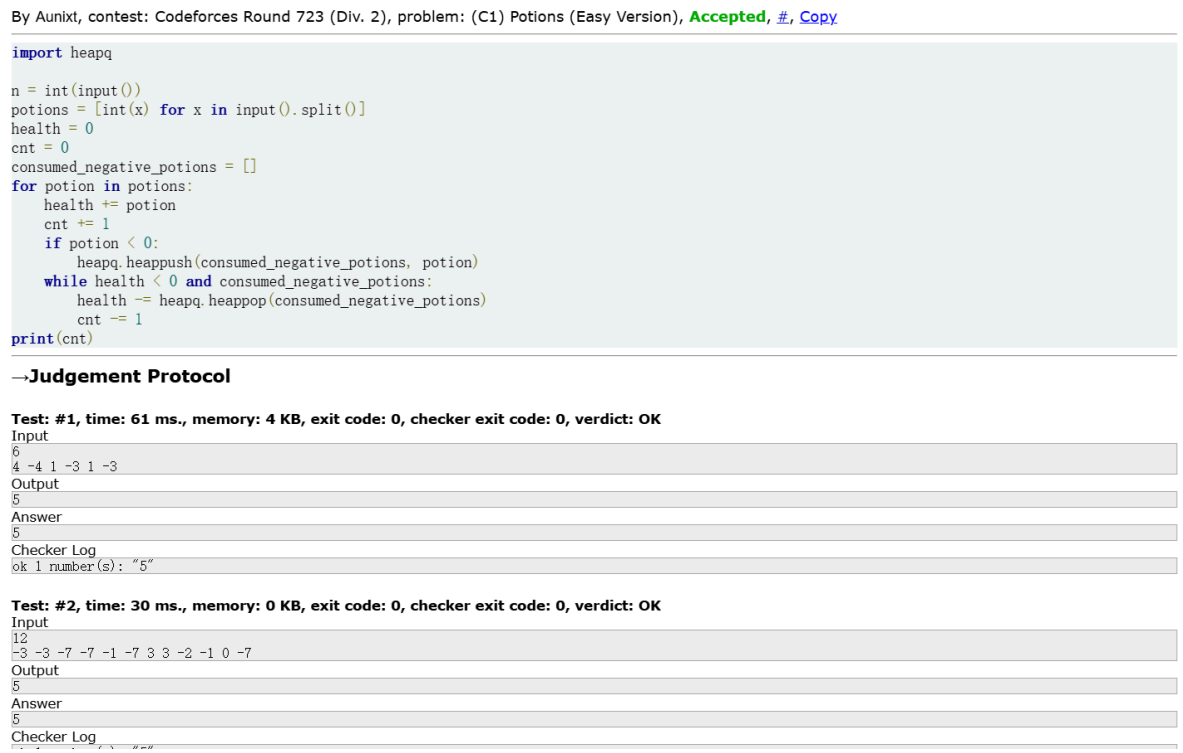
代码:

```
import heapq

n = int(input())
potions = [int(x) for x in input().split()]
health = 0
cnt = 0
consumed_negative_potions = []
for potion in potions:
    health += potion
    cnt += 1
    if potion < 0:
        heapq.heappush(consumed_negative_potions, potion)
    while health < 0 and consumed_negative_potions:
```

```
        health -= heapq.heappop(consumed_negative_potions)
        cnt -= 1
print(cnt)
```

代码运行截图 （至少包含有"Accepted"）



By Aunixt, contest: Codeforces Round 723 (Div. 2), problem: (C1) Potions (Easy Version), **Accepted**, #, Copy

```
import heapq

n = int(input())
potions = [int(x) for x in input().split()]
health = 0
cnt = 0
consumed_negative_potions = []
for potion in potions:
    health += potion
    cnt += 1
    if potion < 0:
        heapq.heappush(consumed_negative_potions, potion)
    while health < 0 and consumed_negative_potions:
        health -= heapq.heappop(consumed_negative_potions)
        cnt -= 1
print(cnt)
```

→**Judgement Protocol**

Test: #1, time: 61 ms., memory: 4 KB, exit code: 0, checker exit code: 0, verdict: OK
Input
```
6
4 -4 1 -3 1 -3
```
Output
```
5
```
Answer
```
5
```
Checker Log
```
ok 1 number(s): "5"
```

Test: #2, time: 30 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK
Input
```
12
-3 -3 -7 -7 -1 -7 3 3 -2 -1 0 -7
```
Output
```
5
```
Answer
```
5
```
Checker Log

22067: 快速堆猪

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

思路：

用heap取出最小猪来减少时间，但问题是stack pop了小猪之后无法在heap里准确把这一头猪pop出去，于是用defaultdict记录这一重量的猪被pop的次数，寻找最小值时再从heap里删除。

代码：

```
import heapq
from collections import defaultdict
stack = []
weight = []
deleted = defaultdict(int)
while True:
    try:
        s = input().split()
        if s[0] == 'pop':
            if stack:
                deleted[stack.pop()] += 1
        elif s[0] == 'min':
            if stack:
                while True:
                    x = heapq.heappop(weight)
```

```
                    if not deleted[x]:
                        heapq.heappush(weight, x)
                        print(x)
                        break
                    deleted[x] -= 1
        else:
            n = int(s[1])
            stack.append(n)
            heapq.heappush(weight, n)
    except EOFError:
        break
```

代码运行截图 （至少包含有"Accepted"）

#47678155提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
from collections import defaultdict
stack = []
weight = []
deleted = defaultdict(int)
while True:
    try:
        s = input().split()
        if s[0] == 'pop':
            if stack:
                deleted[stack.pop()] += 1
        elif s[0] == 'min':
            if stack:
                while True:
                    x = heapq.heappop(weight)
                    if not deleted[x]:
                        heapq.heappush(weight, x)
                        print(x)
                        break
                    deleted[x] -= 1
        else:
            n = int(s[1])
            stack.append(n)
            heapq.heappush(weight, n)
    except EOFError:
        break
```

基本信息

#: 47678155
题目: 22067
提交人: 24n2400011033
内存: 6192kB
时间: 356ms
语言: Python3
提交时间: 2024-12-11 11:36:11

English 帮助 关于

20106: 走山路

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

思路：

用heap保证每次取出的均是距离起始点最近的点，用min_cost记录到达每个点所需要的最小消耗，如果pop出来的地方已经有更小的路径，则直接continue跳过

代码：

```
import heapq

def dijkstra():

    heap = []
    heapq.heappush(heap, (0, start_x, start_y))
```

```
    min_cost = [[float('inf')] * n for _ in range(m)]
    min_cost[start_x][start_y] = 0

    while heap:
        num, x, y = heapq.heappop(heap)

        if num > min_cost[x][y]:
            continue

        if x == end_x and y == end_y:
            return num

        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < m and 0 <= ny < n and matrix[nx][ny] != '#':
                cost = num + abs(int(matrix[nx][ny]) - int(matrix[x][y]))
                if cost < min_cost[nx][ny]:
                    min_cost[nx][ny] = cost
                    heapq.heappush(heap, (cost, nx, ny))

    return 'NO'

m, n, p = map(int, input().split())
matrix = [input().split() for _ in range(m)]
directions = [(1, 0), (0, 1), (-1, 0), (0, -1)]

for _ in range(p):
    start_x, start_y, end_x, end_y = map(int, input().split())
    if matrix[start_x][start_y] == '#' or matrix[end_x][end_y] == '#':
        print('NO')
    else:
        print(dijkstra())
```

代码运行截图（至少包含有"Accepted"）

#47701869提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```python
import heapq

def dijkstra():

    heap = []
    heapq.heappush(heap, (0, start_x, start_y))
    min_cost = [[float('inf')] * n for _ in range(m)]
    min_cost[start_x][start_y] = 0

    while heap:
        num, x, y = heapq.heappop(heap)

        if num > min_cost[x][y]:
            continue

        if x == end_x and y == end_y:
            return num

        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < m and 0 <= ny < n and matrix[nx][ny] != '#':
                cost = num + abs(int(matrix[nx][ny]) - int(matrix[x][y]))
                if cost < min_cost[nx][ny]:
                    min_cost[nx][ny] = cost
                    heapq.heappush(heap, (cost, nx, ny))

    return 'NO'

m, n, p = map(int, input().split())
matrix = [input().split() for _ in range(m)]
directions = [(1, 0), (0, 1), (-1, 0), (0, -1)]

for _ in range(p):
    start_x, start_y, end_x, end_y = map(int, input().split())
    if matrix[start_x][start_y] == '#' or matrix[end_x][end_y] == '#':
        print('NO')
```

基本信息

#: 47701869
题目: 20106
提交人: 24n2400011033
内存: 3920kB
时间: 214ms
语言: Python3
提交时间: 2024-12-12 16:24:37

04129: 变换的迷宫

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

思路:

inq增加迷宫状态的维度，如果在此状态下此前已经到达过当前位置，则说明已经存在更优的路径，直接跳过

代码:

```python
from collections import deque

directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
def bfs(start_x, start_y):
    q = deque([(0, start_x, start_y)])
    inq = {(0, start_x, start_y)}
    while q:
        step, x, y = q.popleft()
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            condition = (step+1) % k
            if 0 <= nx < r and 0 <= ny < c and (condition, nx, ny) not in inq:
                if matrix[nx][ny] == 'E':
                    return step + 1
                if condition == 0 or matrix[nx][ny] != '#':
                    q.append((step+1, nx, ny))
```

```
                    inq.add((condition, nx, ny))
    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))
```

代码运行截图 （至少包含有"Accepted"）

#47704299提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
def bfs(start_x, start_y):
    q = deque([(0, start_x, start_y)])
    inq = {(0, start_x, start_y)}
    while q:
        step, x, y = q.popleft()
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            condition = (step+1) % k
            if 0 <= nx < r and 0 <= ny < c and (condition, nx, ny) not :
                if matrix[nx][ny] == 'E':
                    return step + 1
                if condition == 0 or matrix[nx][ny] != '#':
                    q.append((step+1, nx, ny))
                    inq.add((condition, nx, ny))
    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))
```

基本信息

#: 47704299
题目: 04129
提交人: 24n2400011033
内存: 5088kB
时间: 116ms
语言: Python3
提交时间: 2024-12-12 17:22:39

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。

加深了heap和dijkstra算法的了解，同时对剪枝有了更深的认识，比如最后一题加入了取模后的状态维

要看看前面的讲义、跟上最新的每日选做来准备期末考了