

Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Compiled by 任宇桐 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：

如果看到提示的话感觉确实是简单题目，直接代码实现即可。题目的提示个人感觉没有那么容易想到。如果 $a > 2b$ ，那么先手有2种取法，一定可以使得自己获胜，但是另一种情况就只有一种选择，不知道是否获胜，需要继续判断。

代码：

```
def dfs(a, b, step):  
    if a//b >= 2 or a == b:
```

```

        return step
    else:
        return dfs(b, a-b, (step+1)%2)

while True:
    s = list(map(int, input().split()))
    a = max(s)
    b = min(s)
    if a == 0:
        break
    ans = dfs(a, b, 0)
    if ans == 0:
        print('win')
    else:
        print('lose')

```

代码运行截图（至少包含有"Accepted"）

作者:  stur

结果: Accepted

通过了 10/10个数据

运行时间: 790 ms

运行空间: 36540 KB

提交时间: 34秒前

```

1 def dfs(a, b, step):
2     if a//b >= 2 or a == b:
3         return step
4     else:
5         return dfs(b, a-b, (step+1)%2)
6
7 while True:
8     s = list(map(int, input().split()))
9     a = max(s)
10    b = min(s)
11    if a == 0:
12        break
13    ans = dfs(a, b, 0)
14    if ans == 0:
15        print('win')
16    else:
17        print('lose')

```

25570: 洋葱

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

思路:

一眼看起来题目还是挺简单的，后来发现还没有那么简单，需要单独讨论 $i=1, i=2$ 的情形，需要声明。

代码:

```
n = int(input())
s = []
for _ in range(n):
    s.append(list(map(int, input().split())))
ans = []
i = 0
j = n-1
while True:
    if i == j:
        ans.append(s[i][i])
        break
    elif j == i+1:
        ans.append(s[i][i]+s[i][j]+s[j][i]+s[j][j])
        break
    else:
        ans.append(sum(s[i][i:j+1])+sum(s[k][i] for k in range(i+1, j))+sum(s[j][i:j+1])+sum(s[k][j] for k in range(i+1, j)))
        i += 1
        j -= 1
print(max(ans))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
s = []
for _ in range(n):
    s.append(list(map(int, input().split())))
ans = []
i = 0
j = n-1
while True:
    if i == j:
        ans.append(s[i][i])
        break
    elif j == i+1:
        ans.append(s[i][i]+s[i][j]+s[j][i]+s[j][j])
        break
    else:
        ans.append(sum(s[i][i:j+1])+sum(s[k][i] for k in range(i+1, j)))
        i += 1
        j -= 1
print(max(ans))
```

基本信息

#: 47664468
题目: 25570
提交人: 24n2400011498
内存: 3904kB
时间: 22ms
语言: Python3
提交时间: 2024-12-10 16:31:14

1526C1. Potions(Easy Version)

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

思路:

掌握了优秀的数据结构就是不一样，发现完全相同的代码1525C2也可以AC。

代码:

```
import heapq
n = int(input())
s = list(map(int, input().split()))
dp = [0] * (n+1)
ans = 0
q = []
for i in range(1, n+1):
    heapq.heappush(q, s[i-1])
    dp[i] = dp[i-1] + s[i-1]
    if dp[i] >= 0:
        ans += 1
    else:
        worst_potion = heapq.heappop(q)
        dp[i] -= worst_potion
print(ans)
```

代码运行截图（至少包含有"Accepted"）

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

```
import heapq
n = int(input())
s = list(map(int, input().split()))
dp = [0] * (n+1)
ans = 0
q = []
for i in range(1, n+1):
    heapq.heappush(q, s[i-1])
    dp[i] = dp[i-1] + s[i-1]
    if dp[i] >= 0:
        ans += 1
    else:
        worst_potion = heapq.heappop(q)
        dp[i] -= worst_potion
print(ans)
```

Accepted

22067: 快速堆猪

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

思路:

看到题面这么简单就知道一定有时间限制, 容易想到新开一个列表存储最小值方便查找, 感觉有点像Dec月考的第一题的简化版本。

代码:

```
s = []
stack = []
while True:
    try:
        str = input()
    except EOFError:
        break
    if 'push' in str:
        n = int(str[5:])
        s.append(n)
        if stack:
            stack.append(min(stack[-1], n))
        else:
            stack.append(n)
    elif 'pop' in str:
        if s:
```

```

        s.pop()
        stack.pop()
    elif 'min' in str:
        if s:
            print(stack[-1])

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

s = []
stack = []
while True:
    try:
        str = input()
    except EOFError:
        break
    if 'push' in str:
        n = int(str[5:])
        s.append(n)
        if stack:
            stack.append(min(stack[-1], n))
        else:
            stack.append(n)
    elif 'pop' in str:
        if s:
            s.pop()
            stack.pop()
    elif 'min' in str:
        if s:
            print(stack[-1])

```

基本信息

#: 47492531
 题目: 22067
 提交人: 24n2400011498
 内存: 6648kB
 时间: 298ms
 语言: Python3
 提交时间: 2024-12-01 11:42:58

20106: 走山路

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

思路:

看到题目的提示，仔细研究了一下Dijkstra算法的模板，感觉是不断地按照最小值覆盖能到达的点，然后保证最后到达终点时最小，确实有点像bfs，不过bfs是先填充一层，然后填充下一层，Dijkstra需要按照层数排序，从少到多依次完成。

代码:

```

import heapq

def scope(x, y):
    return 0<=x<m and 0<=y<n

```

```

def dijkstra():
    while q:
        od, x, y = heapq.heappop(q)
        for dx, dy in [(0,1), (0, -1), (1,0), (-1,0)]:
            nx = x+dx
            ny = y+dy
            if scope(nx, ny) and maze[nx][ny] != '#':
                d = abs(int(maze[x][y])-int(maze[nx][ny]))
                if distance[nx][ny] > od+d:
                    distance[nx][ny] = od+d
                    heapq.heappush(q, (od+d, nx, ny))

m, n, p = map(int, input().split())
maze = []
for _ in range(m):
    maze.append(list(input().split()))
for _ in range(p):
    a, b, c, d = map(int, input().split())
    if maze[a][b] == '#' or maze[c][d] == '#':
        print('NO')
    else:
        distance = [[float('inf')] * n for _ in range(m)]
        distance[a][b] = 0
        q = [(0, a, b)]
        dijkstra()
        if distance[c][d] == float('inf'):
            print('NO')
        else:
            print(distance[c][d])

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
import heapq

def scope(x, y):
    return 0<=x<m and 0<=y<n

def dijkstra():
    while q:
        od, x, y = heapq.heappop(q)
        for dx, dy in [(0,1), (0,-1), (1,0), (-1,0)]:
            nx = x+dx
            ny = y+dy
            if scope(nx, ny) and maze[nx][ny] != '#':
                d = abs(int(maze[x][y]) - int(maze[nx][ny]))
                if distance[nx][ny] > od+d:
                    distance[nx][ny] = od+d
                    heapq.heappush(q, (od+d, nx, ny))

m, n, p = map(int, input().split())
maze = []
for _ in range(m):
    maze.append(list(input().split()))
```

基本信息

#: 47628397
题目: 20106
提交人: 24n2400011498
内存: 3712kB
时间: 499ms
语言: Python3
提交时间: 2024-12-08 17:10:17

04129: 变换的迷宫

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

思路:

思考1:

开始研究了半天，没有弄明白为什么WA，后来发现在某些情况下点需要重走，要等石块消失以后处理。

思考2:

在明白上面了思路之后发现还不能AC，大概WA了8次左右才发现关键在于，允许走的位置除了'.'，对应时间下的'#'，还有'S'位置也需要重走，感觉坑点太多，考试的时候肯定做不出来...

代码:

```
from collections import deque

def scope(x, y):
    return 0<=x<r and 0<=y<c

def bfs(a, b):
    q = deque([(0, a, b)])
```



```

inqueue = {0, a, b}
while q:
    step, x, y = q.popleft()
    for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
        nx = x + dx
        ny = y + dy
        if scope(nx, ny):
            if maze[nx][ny] == 'E':
                return step + 1
            if ((step+1)%k, nx, ny) not in inqueue:
                if maze[nx][ny] != '#' or (step + 1) % k ==
0:
                    q.append((step+1, nx, ny))
                    inqueue.add(((step+1)%k, nx, ny))

    return 'Oop!'

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

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque

def scope(x, y):
    return 0<=x<r and 0<=y<c

def bfs(a, b):
    q = deque([(0, a, b)])
    inqueue = {0, a, b}
    while q:
        step, x, y = q.popleft()
        for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
            nx = x + dx
            ny = y + dy
            if scope(nx, ny):
                if maze[nx][ny] == 'E':
                    return step + 1
                if ((step+1)%k, nx, ny) not in inqueue:
```

基本信息

#: 47683215
题目: 04129
提交人: 24n2400011498
内存: 4956kB
时间: 134ms
语言: Python3
提交时间: 2024-12-11 16:25:50

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。

感觉做作业确实是五味杂陈...感觉第一题如果没有见过大概率想不到思路，但是发现通过努力还是能够想清楚Potions之类的问题的，bfs代码不仅长，而且变式以后实在难以想出来错点，感觉按照现在自己的水平期末上机不得不开始时就合理放弃几道题目了...