Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 佟永鑫 元培学院

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：

代码：

```
def qushizi(a, b):
    if a < b:
        a, b = b, a
    if b == 0:
        return False
    if a // b >= 2:
        return True
    return not qushizi(b, a - b)

while True:
    a, b = map(int, input().split())
    if a == 0 and b == 0:
        break
    if qushizi(a, b):
        print("win")
    else:
        print("lose")
```

代码运行截图 （至少包含有"Accepted"）

25570: 洋葱

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

思路：

代码：

```
def yangcong(n, yc):
    max_sum = 0
    a = (n + 1) // 2
    for k in range(a):
        current_sum = 0
        for j in range(k, n - k):
            current_sum += yc[k][j]
        for i in range(k + 1, n - k):
            current_sum += yc[i][n - k - 1]
        for j in range(n - k - 2, k - 1, -1):
            current_sum += yc[n - k - 1][j]
        for i in range(n - k - 2, k, -1):
            current_sum += yc[i][k]
        max_sum = max(max_sum, current_sum)
    return max_sum

n = int(input())
yc = [list(map(int, input().split())) for _ in range(n)]
print(yangcong(n, yc))
```

状态：Accepted

源代码

```
def yangcong(n, yc):
    max_sum = 0
    a = (n + 1) // 2
    for k in range(a):
        current_sum = 0
        for j in range(k, n - k):
            current_sum += yc[k][j]
        for i in range(k + 1, n - k):
            current_sum += yc[i][n - k - 1]
        for j in range(n - k - 2, k - 1, -1):
            current_sum += yc[n - k - 1][j]
        for i in range(n - k - 2, k, -1):
            current_sum += yc[i][k]
        max_sum = max(max_sum, current_sum)
    return max_sum

n = int(input())
yc = [list(map(int, input().split())) for _ in range(n)]
print(yangcong(n, yc))
```

代码运行截图 ==（至少包含有"Accepted"）==

1526C1. Potions(Easy Version)

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

思路：

注意只能按顺序喝或不喝，不能回头

代码：

```
import heapq
def max_potions(n, potions):
    health = 0
    count = 0
    heap = []
    for potion in potions:
        if health + potion >= 0:
            health += potion
            heapq.heappush(heap, potion)
            count += 1
        elif heap and potion > heap[0]:
            health += potion - heap[0]
            heapq.heappop(heap)
            heapq.heappush(heap, potion)
    return count

n = int(input())
potions = list(map(int, input().split()))
```

```
print(max_potions(n, potions))
```

代码运行截图 （至少包含有"Accepted"）

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

```
import heapq
def max_potions(n, potions):
    health = 0
    count = 0
    heap = []
    for potion in potions:
        if health + potion >= 0:
            health += potion
            heapq.heappush(heap, potion)
            count += 1
        elif heap and potion > heap[0]:
            health += potion - heap[0]
            heapq.heappop(heap)
            heapq.heappush(heap, potion)
    return count

n = int(input())
potions = list(map(int, input().split()))
print(max_potions(n, potions))
```

22067: 快速堆猪

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

思路：

代码：

```
import sys
input = sys.stdin.read

def pig(operations):
    pigs = []
    min_pig = []
    result = []

    for operation in operations:
        if operation.startswith("push"):
            _, n = operation.split()
            n = int(n)
            pigs.append(n)
            if min_pig:
                min_pig.append(min(n, min_pig[-1]))
            else:
                min_pig.append(n)
        elif operation == "pop":
            if pigs:
                pigs.pop()
                min_pig.pop()

        elif operation == "min":
            if min_pig:
                result.append(str(min_pig[-1]))
```

```
    return "\n".join(result)

data = input().strip().splitlines()
print(pig(data))
```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```
import sys
input = sys.stdin.read

def pig(operations):
    pigs = []
    min_pig = []
    result = []

    for operation in operations:
        if operation.startswith("push"):
            _, n = operation.split()
            n = int(n)
            pigs.append(n)
            if min_pig:
                min_pig.append(min(n, min_pig[-1]))
            else:
                min_pig.append(n)
        elif operation == "pop":
            if pigs:
                pigs.pop()
                min_pig.pop()

        elif operation == "min":
            if min_pig:
                result.append(str(min_pig[-1]))

    return "\n".join(result)

data = input().strip().splitlines()
print(pig(data))
```

20106: 走山路

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

思路:

代码:

```
import heapq
def dijkstra(m, n, Map, start, end):
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
```

```
    if Map[start[0]][start[1]] == "#" or Map[end[0]][end[1]] == "#":
        return "NO"
    INF = float('inf')
    cost = [[INF] * n for _ in range(m)]
    cost[start[0]][start[1]] = 0
    pq = [(0, start[0], start[1])]
    while pq:
        current_cost, x, y = heapq.heappop(pq)
        if (x, y) == end:
            return current_cost
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < m and 0 <= ny < n and Map[nx][ny] != "#":
                new_cost = current_cost + abs(int(Map[x][y]) - int(Map[nx][ny]))

                if new_cost < cost[nx][ny]:
                    cost[nx][ny] = new_cost
                    heapq.heappush(pq, (new_cost, nx, ny))
    return "NO"

def main():
    m, n, p = map(int, input().split())
    Map = []
    for _ in range(m):
        Map.append(input().split())
    for _ in range(p):
        start_x, start_y, end_x, end_y = map(int, input().split())
        start = (start_x, start_y)
        end = (end_x, end_y)
        result = dijkstra(m, n, Map, start, end)
        print(result)
if __name__ == "__main__":
    main()
```

代码运行截图 （至少包含有"Accepted"）

#47796108提交状态

状态: Accepted

源代码

```
import heapq
def dijkstra(m, n, Map, start, end):
    directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]
    if Map[start[0]][start[1]] == "#" or Map[end[0]][end[1]] == "#":
        return "NO"
    INF = float('inf')
    cost = [[INF] * n for _ in range(m)]
    cost[start[0]][start[1]] = 0
    pq = [(0, start[0], start[1])]
    while pq:
        current_cost, x, y = heapq.heappop(pq)
        if (x, y) == end:
            return current_cost
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < m and 0 <= ny < n and Map[nx][ny] != "#":
                new_cost = current_cost + abs(int(Map[x][y]) - int(Map[
                if new_cost < cost[nx][ny]:
                    cost[nx][ny] = new_cost
                    heapq.heappush(pq, (new_cost, nx, ny))
    return "NO"

def main():
    m, n, p = map(int, input().split())
    Map = []
    for _ in range(m):
        Map.append(input().split())
    for _ in range(p):
        start_x, start_y, end_x, end_y = map(int, input().split())
        start = (start_x, start_y)
        end = (end_x, end_y)
        result = dijkstra(m, n, Map, start, end)
        print(result)
if __name__ == "__main__":
    main()
```

04129: 变换的迷宫

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

思路:

代码:

```
from collections import deque
directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

def bfs(R, C, K, Map, start, end):
    queue = deque([(start[0], start[1], 0)])
    visited = [[[False] * K for _ in range(C)] for _ in range(R)]
```

```
    visited[start[0]][start[1]][0] = True

    while queue:
        x, y, time = queue.popleft()
        if (x, y) == end:
            return time
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            next_time = (time + 1) % K
            if 0 <= nx < R and 0 <= ny < C and not visited[nx][ny][next_time]:
                if Map[nx][ny] != '#':
                    visited[nx][ny][next_time] = True
                    queue.append((nx, ny, time + 1))
                elif (time + 1) % K == 0:
                    visited[nx][ny][next_time] = True
                    queue.append((nx, ny, time + 1))

    return "Oop!"

def main():
    T = int(input())
    for _ in range(T):
        R, C, K = map(int, input().split())
        Map = [input().strip() for _ in range(R)]
        start = None
        end = None
        for i in range(R):
            for j in range(C):
                if Map[i][j] == 'S':
                    start = (i, j)
                elif Map[i][j] == 'E':
                    end = (i, j)
        result = bfs(R, C, K, Map, start, end)
        print(result)
if __name__ == "__main__":
    main()
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
from collections import deque
directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

def bfs(R, C, K, Map, start, end):
    queue = deque([(start[0], start[1], 0)])
    visited = [[[False] * K for _ in range(C)] for _ in range(R)]
    visited[start[0]][start[1]][0] = True

    while queue:
        x, y, time = queue.popleft()
        if (x, y) == end:
            return time
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            next_time = (time + 1) % K
            if 0 <= nx < R and 0 <= ny < C and not visited[nx][ny][next_
                if Map[nx][ny] != '#':
                    visited[nx][ny][next_time] = True
                    queue.append((nx, ny, time + 1))
                elif (time + 1) % K == 0:
                    visited[nx][ny][next_time] = True
                    queue.append((nx, ny, time + 1))

    return "Oop!"

def main():
    T = int(input())
    for _ in range(T):
        R, C, K = map(int, input().split())
        Map = [input().strip() for _ in range(R)]
        start = None
        end = None
        for i in range(R):
            for j in range(C):
                if Map[i][j] == 'S':
                    start = (i, j)
                elif Map[i][j] == 'E':
                    end = (i, j)
        result = bfs(R, C, K, Map, start, end)
        print(result)
if __name__ == "__main__":
    main()
```

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。

比月考简单多了。。