Assignment #3: 惊蛰 Mock Exam
Updated 1641 GMT+8 Mar 5, 2025
2025 spring, Complied by 李振硕、信息管理系

1. 题目
E04015: 邮箱验证
strings, http://cs101.openjudge.cn/practice/04015
思路：
代码：

#48512652提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def email_T(email):
    if email[0]=='@' or email[0]=='.' or email[-1]=='.' or email[-1]=='@'
        return 'NO'
    elif email.count('@')!=1:
        return 'NO'

    else:
        for i in range(len(email)-1):
            if email[i]=='@':
                if email[i-1]=='.' or email[i+1]=='.' or  '.' not in emai
                    return 'NO'
                else:
                    return 'YES'


while True:
    try:
        email=input().strip()
        print(email_T(email))
    except EOFError:
        break
```

基本信息
#: 48512652
题目: 04015
提交人: 24n2300093007
内存: 3616kB
时间: 29ms
语言: Python3
提交时间: 2025-03-10 16:56:56

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

```

### M02039: 反反复复

implementation, http://cs101.openjudge.cn/practice/02039/
思路：
代码：

#48514076提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
info = input().strip()
rows = len(info) // n

matrix = []
index = 0
for i in range(rows):
    if i % 2 == 0: # 偶数行，顺序填充
        matrix.append(list(info[index:index + n]))
    else: # 奇数行，逆序填充
        matrix.append(list(info[index:index + n][::-1]))
    index += n

original_message = []
for col in range(n):
    for row in range(rows):
        original_message.append(matrix[row][col])

print("".join(original_message))
```

基本信息
#: 48514076
题目: 02039
提交人: 24n2300093007
内存: 3648kB
时间: 28ms
语言: Python3
提交时间: 2025-03-10 19:06:17

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于
```

M02092: Grandpa is Famous
implementation, http://cs101.openjudge.cn/practice/02092/

思路：

代码：

#48514599提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
while True:
    N,M=map(int,input().split())
    players=[]
    ranking=[]
    last_rank=[]
    total_r=[]
    if N==M==0:
        break
    for i in range(N):
        rank=list(map(int,input().split()))
        for j in range(M):
            if rank[j] not in players:
                players.append(rank[j])
        ranking.append(rank)

    for j in range(len(players)):
        player=players[j]
        totals=0
        for i in range(N):
           total=ranking[i].count(player)
           totals+=total
        if totals not in total_r:
            total_r.append(totals)
        last_rank.append((player,totals))
    last_rank=sorted(last_rank,key=lambda x:x[0])
    total_r=sorted(total_r,reverse=True)
    second_score=total_r[1]
    for i in range(len(players)):
        if last_rank[i][1]==second_score:
            print(last_rank[i][0],end=' ')
    print()
```

基本信息

#: 48514599
题目: 02092
提交人: 24n2300093007
内存: 12508kB
时间: 5730ms
语言: Python3
提交时间: 2025-03-10 19:57:11

M04133: 垃圾炸弹
matrices, http://cs101.openjudge.cn/practice/04133/

思路：

代码：

#48519922提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
d = int(input())
n = int(input())

data = []
for _ in range(n):
    x, y, i = map(int, input().split())
    data.append((x, y, i))

max_garbage = 0
best_locations = []

for i1 in range(1025):
    for i2 in range(1025):
        all_trash = 0


        for x, y, garbage in data:
            if abs(x - i1) <= d and abs(y - i2) <= d:
                all_trash += garbage

        if all_trash > max_garbage:
            max_garbage = all_trash
            best_locations = [(i1, i2)]
        elif all_trash == max_garbage:
            best_locations.append((i1, i2))

print(len(best_locations), max_garbage)
```

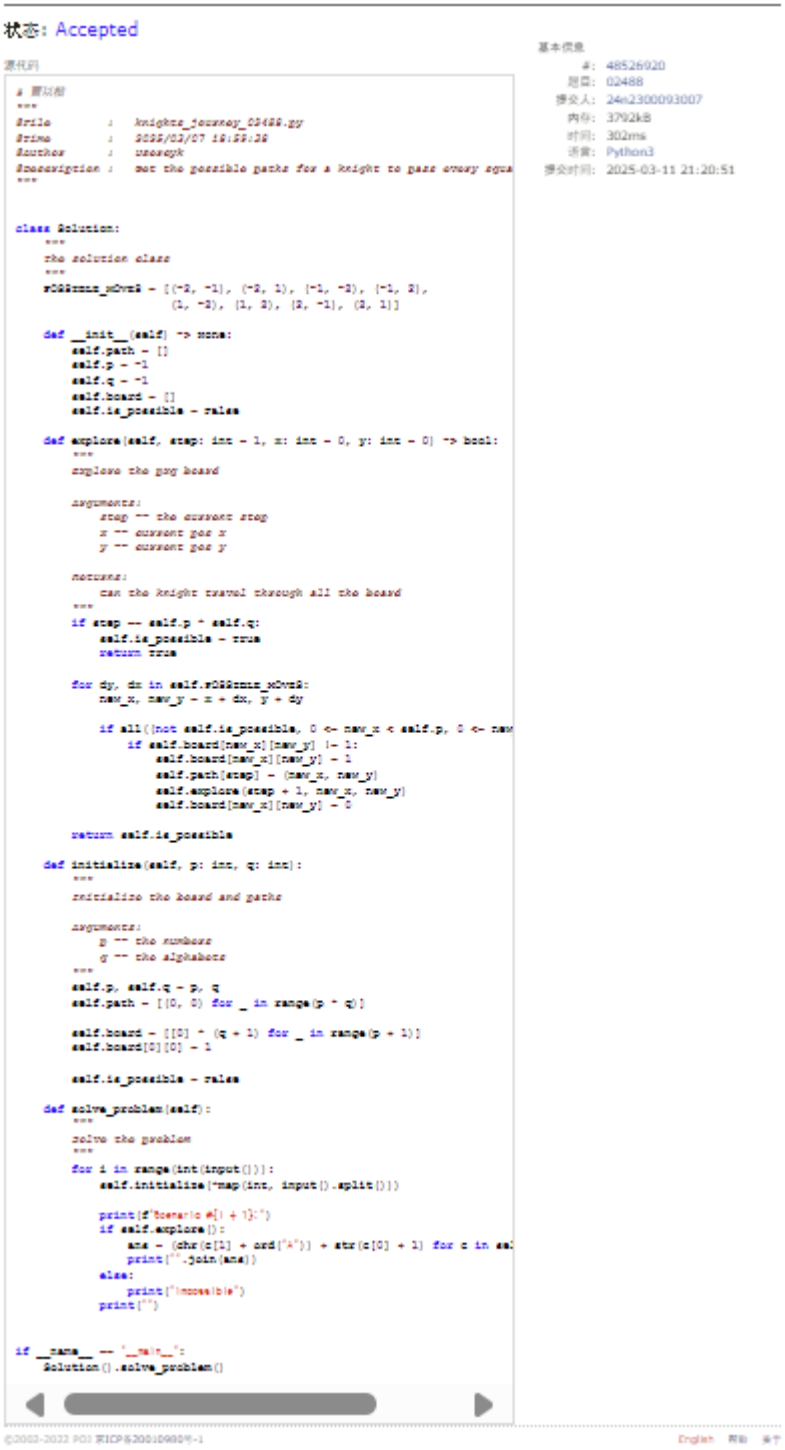
基本信息

#: 48519922
题目: 04133
提交人: 24n2300093007
内存: 53540kB
时间: 1060ms
语言: Python3
提交时间: 2025-03-11 14:38:52

T02488: A Knight's Journey
backtracking, http://cs101.openjudge.cn/practice/02488/

思路：

代码：


状态: Accepted

基本信息

#: 48526020
题目: 02488
提交人: 24n2300093007
内存: 3792kB
时间: 302ms
语言: Python3
提交时间: 2025-03-11 21:20:51

源代码

```
# 导入模块
"""
#file        :   knights_journey_02488.py
#time        :   2025/03/07 16:59:36
#author      :   usaeygk
#description :   get the possible paths for a knight to pass every squa
"""


class Solution:
    """
    the solution class
    """
    self.dxdy = [(-2, -1), (-2, 1), (-1, -2), (-1, 2),
                 (1, -2), (1, 2), (2, -1), (2, 1)]

    def __init__(self) -> none:
        self.path = []
        self.p = -1
        self.q = -1
        self.board = []
        self.is_possible = false

    def explore(self, step: int = 1, x: int = 0, y: int = 0) -> bool:
        """
        explore the new board

        arguments:
            step -- the current step
            x -- current pos x
            y -- current pos y

        returns:
            can the knight travel through all the board
        """
        if step == self.p * self.q:
            self.is_possible = true
            return true

        for dy, dx in self.dxdy:
            new_x, new_y = x + dx, y + dy

            if all((not self.is_possible, 0 <= new_x < self.p, 0 <= new
                if self.board[new_x][new_y] != 1:
                    self.board[new_x][new_y] = 1
                    self.path[step] = (new_x, new_y)
                    self.explore(step + 1, new_x, new_y)
                    self.board[new_x][new_y] = 0

        return self.is_possible

    def initialize(self, p: int, q: int):
        """
        initialize the board and paths

        arguments:
            p -- the numbers
            q -- the alphabets
        """
        self.p, self.q = p, q
        self.path = [(0, 0) for _ in range(p * q)]

        self.board = [[0] * (q + 1) for _ in range(p + 1)]
        self.board[0][0] = 1

        self.is_possible = false

    def solve_problem(self):
        """
        solve the problem
        """
        for i in range(int(input())):
            self.initialize(*map(int, input().split()))

            print(f"Scenario #{i + 1}:")
            if self.explore():
                ans = (chr(c[1] + ord("A")) + str(c[0] + 1) for c in sel
                print("".join(ans))
            else:
                print("impossible")
            print("")


if __name__ == "__main__":
    Solution().solve_problem()
```

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

T06648: Sequence
heap, http://cs101.openjudge.cn/practice/06648/

思路：

代码：

#48526976提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq

def find_min_sums():
    test_cases = int(input())
    for _ in range(test_cases):
        num_sequences, sequence_length = map(int, input().split())
        first_sequence = sorted(map(int, input().split()))

        for _ in range(num_sequences - 1):
            next_sequence = sorted(map(int, input().split()))

            min_heap = [(first_sequence[i] + next_sequence[0], i, 0) fo
            heapq.heapify(min_heap)

            result = []
            for _ in range(sequence_length):
                current_sum, i, j = heapq.heappop(min_heap)
                result.append(current_sum)

                if j + 1 < len(next_sequence):
                    heapq.heappush(min_heap, (first_sequence[i] + next_

            first_sequence = result

        print(*first_sequence)

if __name__ == "__main__":
    find_min_sums()
```

基本信息

#: 48526976
题目: 06648
提交人: 24n2300093007
内存: 7508kB
时间: 850ms
语言: Python3
提交时间: 2025-03-11 21:23:36

2．学习总结和收获

这次月考有些题是之前做过的题，但还是不会，感觉需要复习上学期的内容。