

第3次作業-作業-HW3

學號：112111215

姓名：莊博勳

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2024/11/24

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

1. 請回答下面問題。

Ans:

```
def isFull(Stack,top,N):  
    if top==N-1: #如果top指標指向堆疊頂端,傳回True  
        return True  
    else: #否則傳回False  
        return False  
  
def isEmpty(stack,top):  
    if top==-1: #如果top指標為-1,傳回True  
        return True  
    else: #否則傳回False  
        return False
```

2. 請回答下面問題。

Ans:

```
def is_valid_move(x, y, N, visited): #檢查是否在棋盤內，且該格子未被訪問  
    return 0 <= x < N and 0 <= y < N and not visited[x][y]  
  
def knight_tour(N, startX, startY): #騎士的8個移動方向  
    moves = [(-2, -1), (-2, 1), (-1, -2), (-1, 2), (1, -2), (1, 2), (2, -1), (2, 1)]  
    visited = [[False] * N for _ in range(N)] #初始化棋盤的訪問記錄  
    visited_count = 0 #記錄已訪問的格子數量  
  
    def dfs(x, y, visited_count):  
        visited[x][y] = True #當前格子被訪問，將其標記為已訪問  
        visited_count += 1  
        if visited_count == N * N:  
            return True #如果已經訪問了所有格子，返回 True  
        for move in moves: #嘗試每個可能的移動方向  
            nx, ny = x + move[0], y + move[1]
```

```
def is_valid_move(nx, ny, N, visited):
    if dfs(nx, ny, visited_count): #如果能夠移動到(nx,ny)，則繼續進行DFS
        return True
    visited[x][y] = False #如果沒有可行的移動，則回溯
    return False
return dfs(startX, startY, visited_count) #開始從起始位置進行DFS

N = int(input()) #棋盤大小
startX, startY = map(int, input().split()) #起始位置

if knight_tour(N, startX, startY): #呼叫函數並輸出結果
    print(True)
else:
    print(False)
```

3. 請回答下面問題：

Ans:

```
def josephus(n, k):
    people = list(range(1, n + 1)) #用列表來表示所有人，列表中的每個元素對應每個人的編號
    index = 0 #記錄當前需要刪除的位置
    while len(people) > 1: #直到只剩下1個人
        index = (index + k - 1) % len(people) #計算應該刪除的人位置，這是循環計數
        people.pop(index) #刪除該人
    return people[0] #返回最後剩下的人的編號

n, k = map(int, input().split())
print(josephus(n, k)) #輸出最後剩下的人的編號
```

個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結