Assignment #5: cs201 Mock Exam寒露第三天

Updated 1913 GMT+8 Oct 10, 2025

2025 fall, Complied by 杨知进 物理学院

> **说明：**
>
> 1. **解题与记录：**
>
> 对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E29952: 咒语序列

Stack, http://cs101.openjudge.cn/practice/29952/

思路：

一开始理解错了题目，以为必须在一个（()）里，不能（）（）。用enumerate迭代；用栈存储数据，当前索引-栈中索引为当前匹配长度

代码：

```
s = input()
stack = [-1]
maxout = 0
for i, char in enumerate(s):
    if char == '(':
        stack.append(i)
```

```
    else:
        stack.pop()
        if not stack:
            stack.append(i)
        else:
            count = i - stack[-1]
            if count > maxout:
                maxout = count
print(maxout)
```

代码运行截图（至少包含有"Accepted"）

#50327714提交状态

查看 提交 统计 提

状态：Accepted

源代码

```
s = input()
stack = [-1]
maxout = 0
for i, char in enumerate(s):
    if char == '(':
        stack.append(i)
    else:
        stack.pop()
        if not stack:
            stack.append(i)
        else:
            count = i - stack[-1]
            if count > maxout:
                maxout = count
print(maxout)
```

基本信息

#:	50327714
题目:	29952
提交人:	25n2400011575
内存:	4188kB
时间:	26ms
语言:	Python3
提交时间:	2025-10-12 17:41:29

用时：30min左右

M01328: Radar Installation

greedy, http://cs101.openjudge.cn/practice/01328/

思路：

while循环读取输入，注意细节处理。二维坐标的islands实质上转化为数轴上的有效区间sections。贪心算法：最左端开始每次只关注局部，在最极限的位置安装雷达，然后往右看。

代码：

```
case = 1
while True:
    n, d = map(int, input().split())
    if n == 0 and d == 0:
        break
    islands = []
```

```python
    sol = True
    for _ in range(n):
        x, y = map(int, input().split())
        islands.append((x, y))
        if y > d or d < 0:
            sol = False
    junk = input()
    if not sol:
        print(f"Case {case}: -1")
        case+= 1
        continue
    if n == 1 and d>0:
        print(f'Case {case}: 1')
        case+=1
        continue
    sections = []
    for i in islands:
        dx = (d**2 - i[1]**2)**0.5
        left = i[0] - dx
        right = i[0] + dx
        sections.append((left, right))
        sections.sort(key=lambda j: j[1])
    count = 1
    position = sections[0][1]
    for left, right in sections:
        if position < left:
            count += 1
            position = right
    print(f"Case {case}: {count}")
    case+=1
```

代码运行截图（至少包含有"Accepted"）

#50328647提交状态

查看　提交　统计　提问

状态：Accepted

源代码

```python
case = 1
while True:
    n, d = map(int, input().split())
    if n == 0 and d == 0:
        break
    islands = []
    sol = True
    for _ in range(n):
        x, y = map(int, input().split())
```

基本信息

#:	50328647
题目:	01328
提交人:	25n2400011575
内存:	3836kB
时间:	239ms
语言:	Python3
提交时间:	2025-10-12 18:53:56

用时：20min左右

M02754: 八皇后

dfs, http://cs101.openjudge.cn/practice/02754/

思路：
dfs，考试的时候没有用两个参数x、num，而是用了一个，结果搜索一次后就固定了集合x，没有重置的意识！

代码：

```
def dfs(x,num):
    if num==8:
        sol.append(''.join([str(i) for i in x]))
        return
    for i in range(8):
        for j in range(num):
            if i+1 == x[j] or num-j == abs(1+i-x[j]):
                break
        else:
            x[num]= i+1
            dfs(x,num+1)
sol = []
dfs([None]*8,0)
n = int(input())
for i in range(n):
    print(sol[int(input())-1])
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
def dfs(x,num):
    if num==8:
        sol.append(''.join([str(i) for i in x]))
        return
    for i in range(8):
        for j in range(num):
            if i+1 == x[j] or num-j == abs(1+i-x[j]):
```

基本信息

#:	50333212
题目:	02754
提交人:	25n2400011575
内存:	3628kB
时间:	33ms
语言:	Python3
提交时间:	2025-10-12 23:41:18

用时：20min左右

M25570: 洋葱

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

思路：
每一层是四个数列求和（上下左右各一个），有固定程序的题目按照要求计算就好。具体求和指

标，举n=3、5的例子并多尝试。

代码：

```
n = int(input())
x=[]
for i in range(n):
    x.append(list(map(int,input().split())))
s = []
if n%2 == 0:
    for i in range(n//2):
        out=0
        if i == n//2-1:
            out=0
            out+=x[n//2][n//2]+x[n//2-1][n//2-1]+x[n//2-1][n//2]+x[n//2][n//2-
1]
            s.append(out)
        else:
            out+=sum(x[i][i:n-i])+sum(x[n-i-1][i:n-i])
            out+=sum(x[j][i] for j in range(i+1,n-i-1))+sum(x[j][n-i-1] for j
in range(i+1,n-i-1))
            s.append(out)
else:
    for i in range((n+1)//2):


        if i == (n+1)//2-1:
            out=0
            out+=x[(n+1)//2-1][(n+1)//2-1]
            s.append(out)

        else:
            out=0
            out+=sum(x[i][i:n-i])+sum(x[n-i-1][i:n-i])
            out+=sum(x[j][i] for j in range(i+1,n-i-1))+sum(x[j][n-i-1] for j
in range(i+1,n-i-1))
            s.append(out)

print(max(s))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
n = int(input())
x=[]
for i in range(n):
    x.append(list(map(int,input().split())))
s = []
if n%2 == 0:
    for i in range(n//2):
        out=0
```

基本信息

#:	50328702
题目:	25570
提交人:	25n2400011575
内存:	3892kB
时间:	21ms
语言:	Python3
提交时间:	2025-10-12 18:57:57

用时：10min左右

M29954: 逃离紫罗兰监狱

bfs, http://cs101.openjudge.cn/practice/29954/

思路：

学习了bfs。

代码

```
r,c,k = map(int,input().split())
land = [input().strip() for i in range(r)]
start,end = 0,0
for i in range(r):
    for j in range(c):
        if land[i][j] == 'S':
            start = (i,j)
        elif land[i][j] == 'E':
            end = (i,j)
dx = [-1,1,0,0]
dy = [0,0,1,-1]
visited = [[[False]*(k+1) for i in range(c)] for j in range(r)]
go = []
visited[start[0]][start[1]][k] = True
go.append((start[0], start[1], k, 0))
def solve(num):
    while num < len(go):
        x,y,nowk,step= go[num]
        num+=1
        if (x,y) == end:
            print(step)
            return
        for i in range(4):
            nx = x+dx[i]
            '''此处一定要新定义变量，不能沿用x+=，否则在遍历中会一直改变。
```

```
            在每个for下面赋值nx、ny才是对的'''
            ny = y+dy[i]
            if 0<=nx<r and 0<=ny<c:
                if land[nx][ny] == '.' or land[nx][ny] == 'E':
                    if not visited[nx][ny][nowk]:
                        visited[nx][ny][nowk] = True
                        go.append((nx,ny,nowk,step+1))
                elif land[nx][ny] == '#' and nowk > 0:
                    if not visited[nx][ny][nowk-1]:
                        visited[nx][ny][nowk-1] = True
                        go.append((nx,ny,nowk-1,step+1))
    print(-1)
solve(0)
```

（至少包含有"Accepted"）

状态：Accepted

源代码

```
r,c,k = map(int,input().split())
land = [input().strip() for i in range(r)]
start,end = 0,0
for i in range(r):
    for j in range(c):
        if land[i][j] == 'S':
            start = (i,j)
        elif land[i][j] == 'E':
            end = (i,j)
```

基本信息

#:	50347236
题目:	29954
提交人:	25n2400011575
内存:	14588kB
时间:	255ms
语言:	Python3
提交时间:	2025-10-13 21:44:47

用时：40min左右

T27256: 当前队列中位数

backtracking, http://cs101.openjudge.cn/practice/27256/

思路：

仍然在学习中.....

代码

```

```

（至少包含有"Accepted"）

用时：∞

2. 学习总结和个人收获

bfs、dfs的基础不牢固，要认真学习。
反复犯循环变量跨循环使用导致问题的错误，要勤于指定新变量不要耍小聪明觉得方便。
各种排序、更快的优化方法、堆等数据结构.....还有非常多知识要补充学习！！！！！