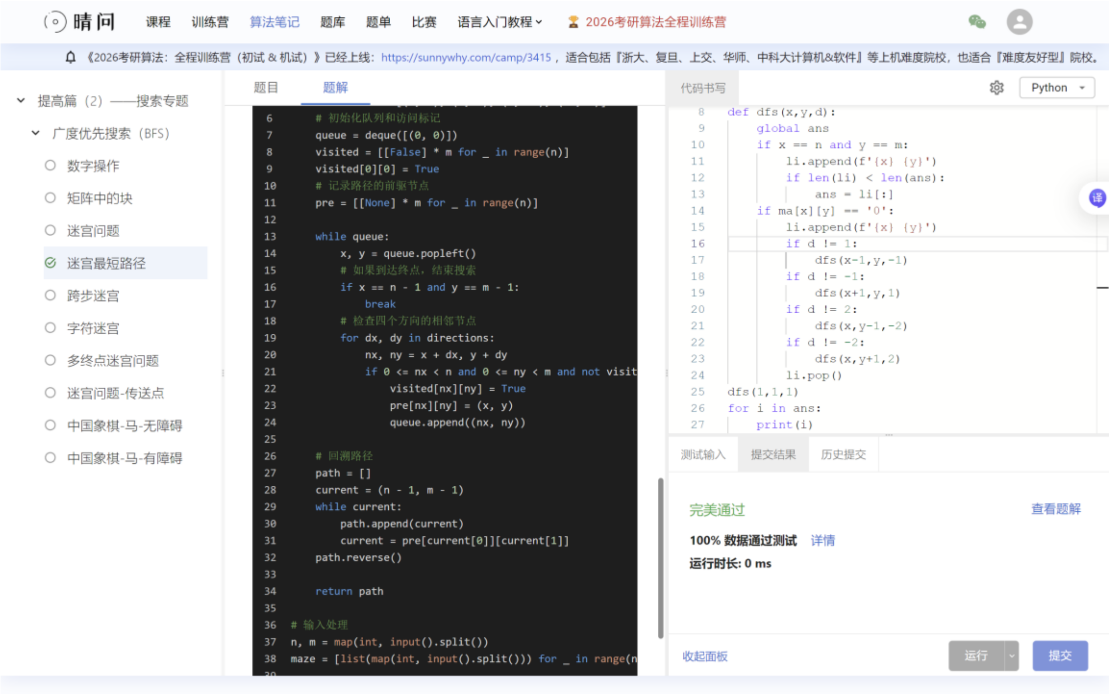
sy321 迷宫最短路径

用 dfs 和 bfs 各写了一遍，想看看两者时间的差距，结果……都是 0ms……



《2026考研算法：全程训练营（初试 & 机试）》已经上线：https://sunnywhy.com/camp/3415，适合包括『浙大、复旦、上交、华师、中科大计算机&软件』等上机难度院校，也适合『难度友好型』院校。

```python
    # 初始化队列和访问标记
    queue = deque([(0, 0)])
    visited = [[False] * m for _ in range(n)]
    visited[0][0] = True
    # 记录路径的前驱节点
    pre = [[None] * m for _ in range(n)]

    while queue:
        x, y = queue.popleft()
        # 如果到达终点，结束搜索
        if x == n - 1 and y == m - 1:
            break
        # 检查四个方向的相邻节点
        for dx, dy in directions:
            nx, ny = x + dx, y + dy
            if 0 <= nx < n and 0 <= ny < m and not visit
                visited[nx][ny] = True
                pre[nx][ny] = (x, y)
                queue.append((nx, ny))

    # 回溯路径
    path = []
    current = (n - 1, m - 1)
    while current:
        path.append(current)
        current = pre[current[0]][current[1]]
    path.reverse()

    return path

# 输入处理
n, m = map(int, input().split())
maze = [list(map(int, input().split())) for _ in range(n
```

```python
def dfs(x,y,d):
    global ans
    if x == n and y == m:
        li.append(f'{x} {y}')
        if len(li) < len(ans):
            ans = li[:]
    if ma[x][y] == '0':
        li.append(f'{x} {y}')
        if d != 1:
            dfs(x-1,y,-1)
        if d != -1:
            dfs(x+1,y,1)
        if d != 2:
            dfs(x,y-1,-2)
        if d != -2:
            dfs(x,y+1,2)
        li.pop()
dfs(1,1,1)
for i in ans:
    print(i)
```

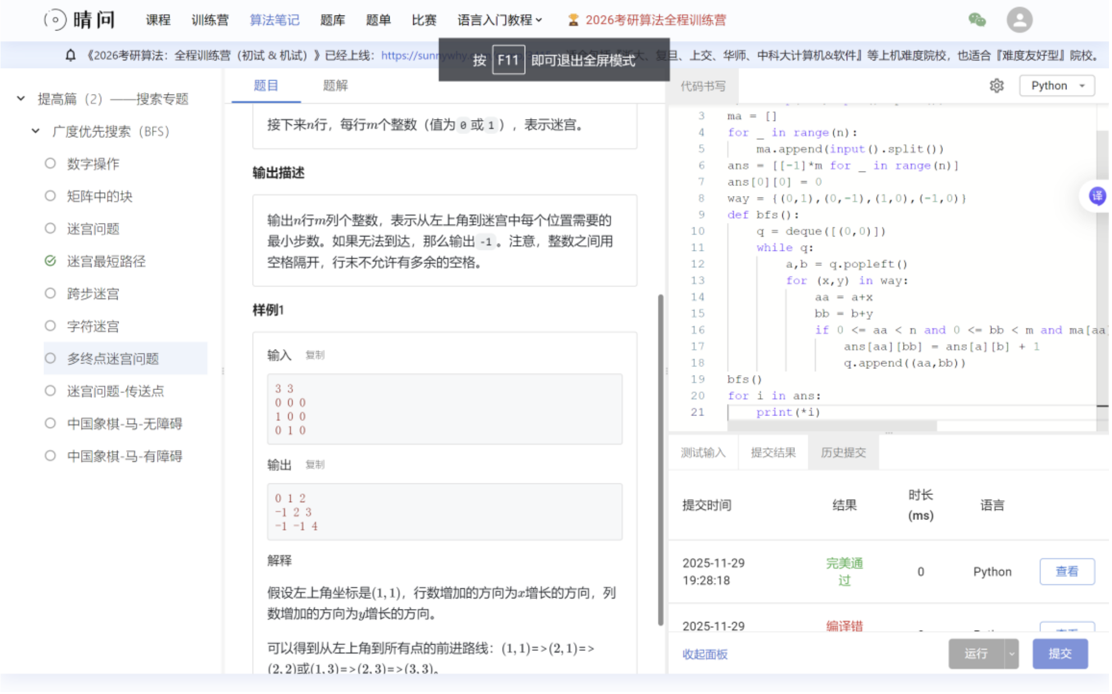
完美通过

100% 数据通过测试

运行时长: 0 ms

sy324 多终点迷宫问题

忘了调用函数，检查了半个多小时才发现……



《2026考研算法：全程训练营（初试 & 机试）》已经上线：https://sunn… …大、复旦、上交、华师、中科大计算机&软件』等上机难度院校，也适合『难度友好型』院校。

接下来n行，每行m个整数（值为 `0` 或 `1` ），表示迷宫。

输出描述

输出n行m列个整数，表示从左上角到迷宫中每个位置需要的最小步数。如果无法到达，那么输出 `-1`。注意，整数之间用空格隔开，行末不允许有多余的空格。

样例1

输入

```
3 3
0 0 0
1 0 0
0 1 0
```

输出

```
0 1 2
-1 2 3
-1 -1 4
```

解释

假设左上角坐标是$(1,1)$，行数增加的方向为x增长的方向，列数增加的方向为y增长的方向。

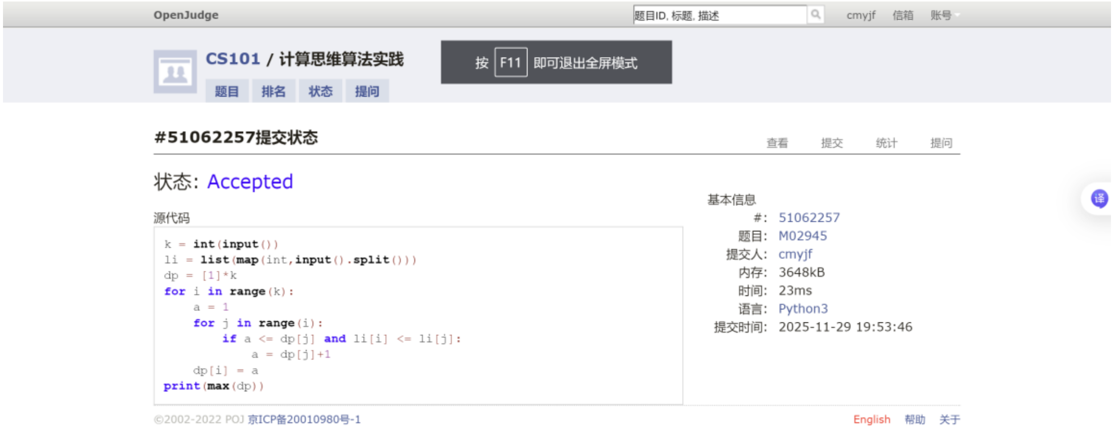
可以得到从左上角到所有点的前进路线：$(1,1)$=>$(2,1)$=>$(2,2)$或$(1,3)$=>$(2,3)$=>$(3,3)$。

```python
ma = []
for _ in range(n):
    ma.append(input().split())
ans = [[-1]*m for _ in range(n)]
ans[0][0] = 0
way = {(0,1),(0,-1),(1,0),(-1,0)}
def bfs():
    q = deque([(0,0)])
    while q:
        a,b = q.popleft()
        for (x,y) in way:
            aa = a+x
            bb = b+y
            if 0 <= aa < n and 0 <= bb < m and ma[aa
                ans[aa][bb] = ans[a][b] + 1
                q.append((aa,bb))
bfs()
for i in ans:
    print(*i)
```

提交时间	结果	时长 (ms)	语言	
2025-11-29 19:28:18	完美通过	0	Python	查看
2025-11-29	编译错			

M02945: 拦截导弹

自己写的 dp，看了题解，有 dp 和二分查找，请问老师，贪心解法是什么？



#51062257提交状态

状态: Accepted

源代码

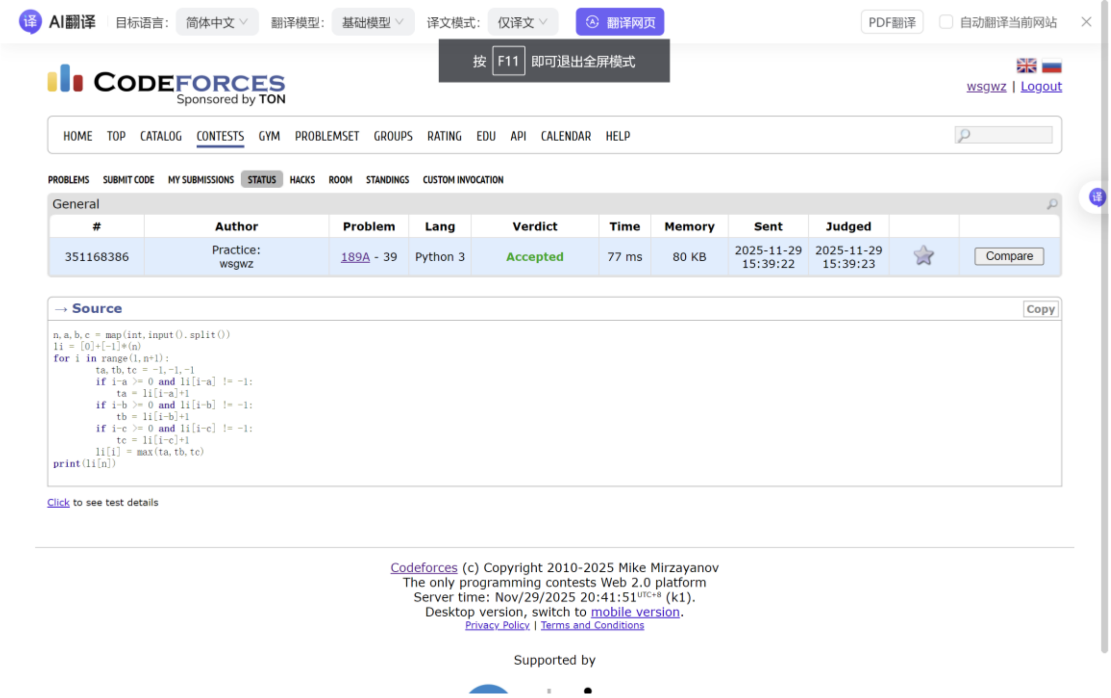
```python
k = int(input())
li = list(map(int,input().split()))
dp = [1]*k
for i in range(k):
    a = 1
    for j in range(i):
        if a <= dp[j] and li[i] <= li[j]:
            a = dp[j]+1
    dp[i] = a
print(max(dp))
```

基本信息

#:	51062257
题目:	M02945
提交人:	cmyjf
内存:	3648kB
时间:	23ms
语言:	Python3
提交时间:	2025-11-29 19:53:46

©2002-2022 POJ 京ICP备20010980号-1

189A. Cut Ribbon



General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged
351168386	Practice: wsgwz	189A - 39	Python 3	Accepted	77 ms	80 KB	2025-11-29 15:39:22	2025-11-29 15:39:23

→ Source

```python
n,a,b,c = map(int,input().split())
li = [0]+[-1]*(n)
for i in range(1,n+1):
        ta,tb,tc = -1,-1,-1
        if i-a >= 0 and li[i-a] != -1:
            ta = li[i-a]+1
        if i-b >= 0 and li[i-b] != -1:
            tb = li[i-b]+1
        if i-c >= 0 and li[i-c] != -1:
            tc = li[i-c]+1
        li[i] = max(ta,tb,tc)
print(li[n])
```

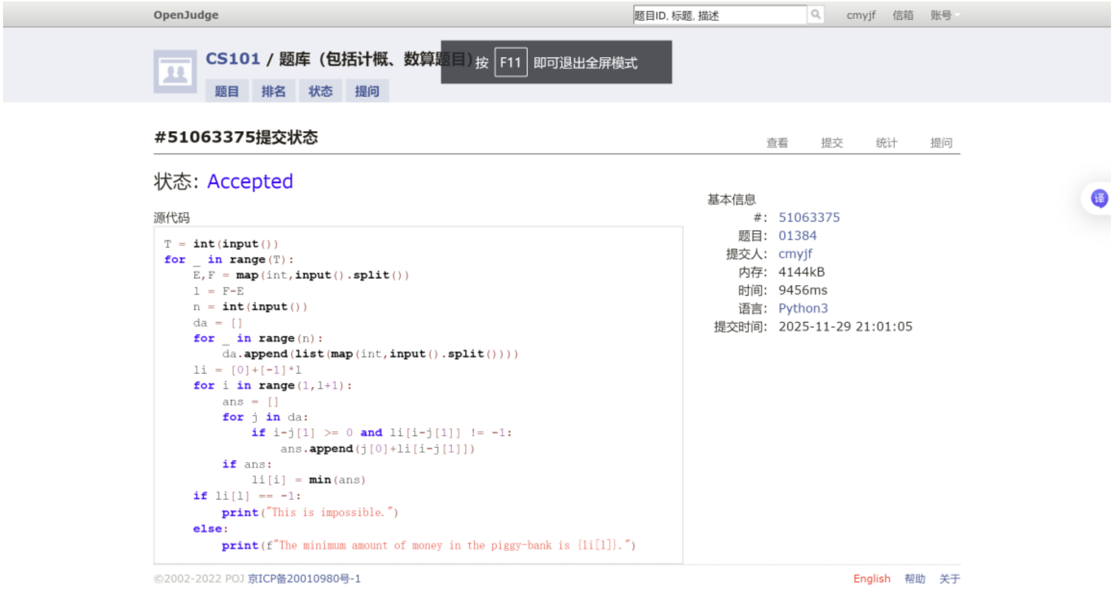
Click to see test details

Codeforces (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Nov/29/2025 20:41:51^{UTC+8} (k1).
Desktop version, switch to mobile version.
Privacy Policy | Terms and Conditions

Supported by

01384:Piggy-Bank

方法和上一题好像，python 惊险通过



#51063375提交状态

状态: Accepted

源代码

```python
T = int(input())
for _ in range(T):
    E,F = map(int,input().split())
    l = F-E
    n = int(input())
    da = []
    for _ in range(n):
        da.append(list(map(int,input().split())))
    li = [0]+[-1]*l
    for i in range(1,l+1):
        ans = []
        for j in da:
            if i-j[1] >= 0 and li[i-j[1]] != -1:
                ans.append(j[0]+li[i-j[1]])
        if ans:
            li[i] = min(ans)
    if li[l] == -1:
        print("This is impossible.")
    else:
        print(f"The minimum amount of money in the piggy-bank is {li[l]}.")
```

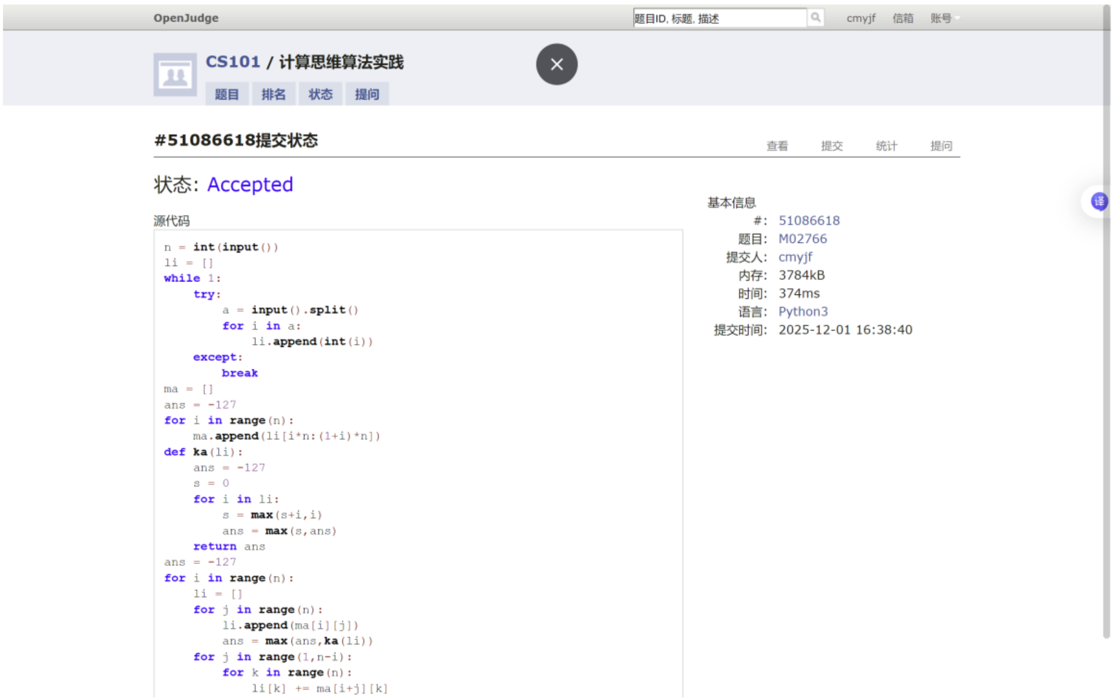
基本信息
#: 51063375
题目: 01384
提交人: cmyjf
内存: 4144kB
时间: 9456ms
语言: Python3
提交时间: 2025-11-29 21:01:05

©2002-2022 POJ 京ICP备20010980号-1

M02766: 最大子矩阵

奇怪的不定行输入

学会了 Kadane 算法



#51086618提交状态

状态: Accepted

源代码

```python
n = int(input())
li = []
while 1:
    try:
        a = input().split()
        for i in a:
            li.append(int(i))
    except:
        break
ma = []
ans = -127
for i in range(n):
    ma.append(li[i*n:(1+i)*n])
def ka(li):
    ans = -127
    s = 0
    for i in li:
        s = max(s+i,i)
        ans = max(s,ans)
    return ans
ans = -127
for i in range(n):
    li = []
    for j in range(n):
        li.append(ma[i][j])
        ans = max(ans,ka(li))
    for j in range(1,n-i):
        for k in range(n):
            li[k] += ma[i+j][k]
```

基本信息
#: 51086618
题目: M02766
提交人: cmyjf
内存: 3784kB
时间: 374ms
语言: Python3
提交时间: 2025-12-01 16:38:40