# Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Complied by 任宇桐 物理学院

**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

### 02692: 假币问题

brute force, http://cs101.openjudge.cn/practice/02692

思路：

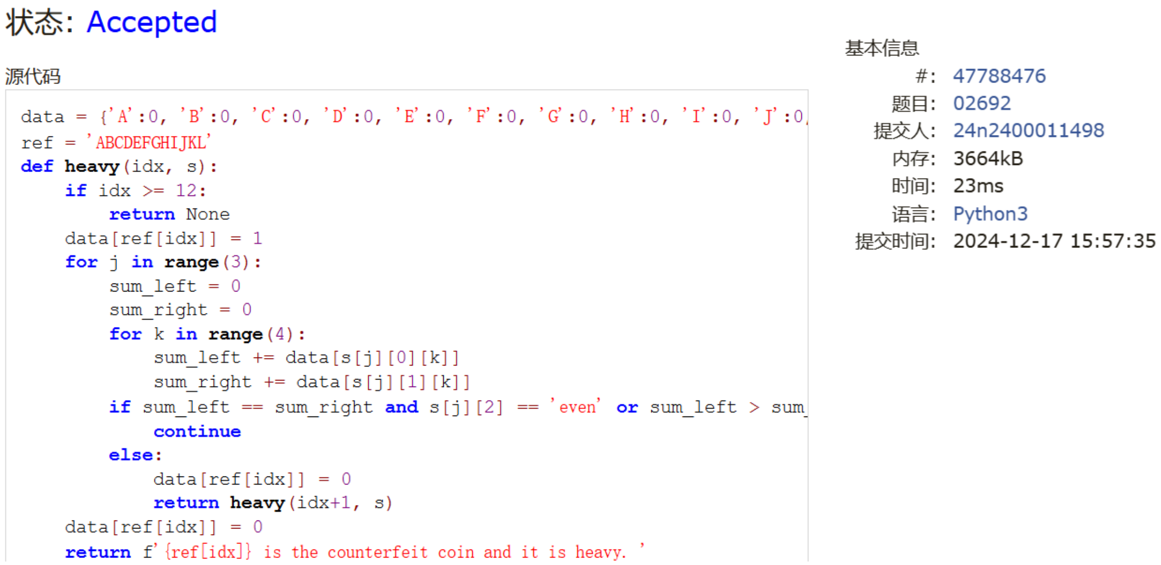
这道题目之前做过，当时就因为返回时没有将初始值设置为0导致错误，现在重新做没想到还是在上面犯了错误，说明检查回溯还是挺重要的。但是话说这道题目如果只需要考虑一个的话应该不用回溯，直接讨论就行...

代码：

```
data = {'A':0, 'B':0, 'C':0, 'D':0, 'E':0, 'F':0, 'G':0, 'H':0,
'I':0, 'J':0, 'K':0, 'L':0}
```

```
ref = 'ABCDEFGHIJKL'
def heavy(idx, s):
    if idx >= 12:
        return None
    data[ref[idx]] = 1
    for j in range(3):
        sum_left = 0
        sum_right = 0
        for k in range(4):
            sum_left += data[s[j][0][k]]
            sum_right += data[s[j][1][k]]
        if sum_left == sum_right and s[j][2] == 'even' or sum_left
> sum_right and s[j][2] == 'up' or sum_left < sum_right and s[j][2]
== 'down':
            continue
        else:
            data[ref[idx]] = 0
            return heavy(idx+1, s)
    data[ref[idx]] = 0
    return f'{ref[idx]} is the counterfeit coin and it is heavy. '


def light(idx, s):
    if idx >= 12:
        return

    data[ref[idx]] = -1
    for j in range(3):
        sum_left = 0
        sum_right = 0
        for k in range(4):
            sum_left += data[s[j][0][k]]
            sum_right += data[s[j][1][k]]
        if sum_left == sum_right and s[j][2] == 'even' or sum_left
> sum_right and s[j][
            2] == 'up' or sum_left < sum_right and s[j][2] ==
'down':
            continue
        else:
            data[ref[idx]] = 0
            return light(idx + 1, s)
    data[ref[idx]] = 0
    return f'{ref[idx]} is the counterfeit coin and it is light. '

t = int(input())
for _ in range(t):
    s = []
    for _ in range(3):
        s.append(list(input().split()))
    if heavy(0, s) is None:
        print(light(0 ,s))
    else:
        print(heavy(0, s))
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
data = {'A':0, 'B':0, 'C':0, 'D':0, 'E':0, 'F':0, 'G':0, 'H':0, 'I':0, 'J':0,
ref = 'ABCDEFGHIJKL'
def heavy(idx, s):
    if idx >= 12:
        return None
    data[ref[idx]] = 1
    for j in range(3):
        sum_left = 0
        sum_right = 0
        for k in range(4):
            sum_left += data[s[j][0][k]]
            sum_right += data[s[j][1][k]]
        if sum_left == sum_right and s[j][2] == 'even' or sum_left > sum_
            continue
        else:
            data[ref[idx]] = 0
            return heavy(idx+1, s)
    data[ref[idx]] = 0
    return f'{ref[idx]} is the counterfeit coin and it is heavy. '
```

基本信息

#: 47788476

题目: 02692

提交人: 24n2400011498

内存: 3664kB

时间: 23ms

语言: Python3

提交时间: 2024-12-17 15:57:35

## 01088: 滑雪

dp, dfs similar, http://cs101.openjudge.cn/practice/01088

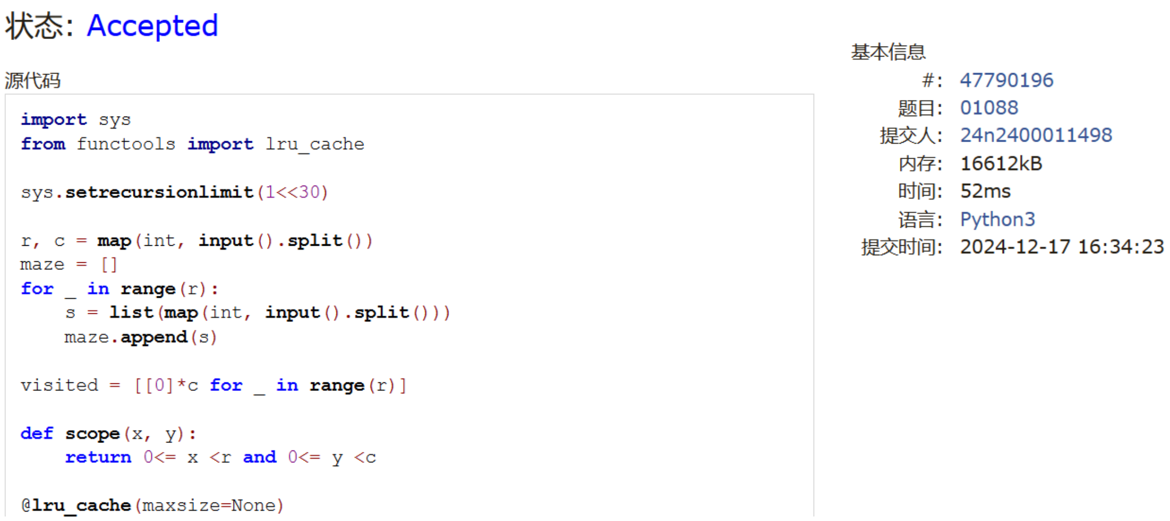
思路：

思路还是很容易通过递归想到的，但是开始时实在没有理解这个滑道是怎么计算的，如果不能滑的话是当成0还是1题目也没有说清楚，只能逐一尝试，希望考试的时候不要出现类似的题目，一但出错就会很紧张...

代码：

```python
import sys
from functools import lru_cache

sys.setrecursionlimit(1<<30)

r, c = map(int, input().split())
maze = []
for _ in range(r):
    s = list(map(int, input().split()))
    maze.append(s)

visited = [[0]*c for _ in range(r)]

def scope(x, y):
    return 0<= x <r and 0<= y <c

@lru_cache(maxsize=None)
def dfs(x, y):
    temp = 0
    for dx, dy in [(0, 1), (0 ,-1), (1, 0), (-1, 0)]:
        nx, ny = x+dx, y+dy
        if scope(nx, ny) and maze[nx][ny] > maze[x][y]:
            temp = max(dfs(nx, ny)+1, temp)
    return temp

ans = 0
for i in range(r):
    for j in range(c):
        ans = max(ans, dfs(i, j))
print(ans+1)
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
import sys
from functools import lru_cache

sys.setrecursionlimit(1<<30)

r, c = map(int, input().split())
maze = []
for _ in range(r):
    s = list(map(int, input().split()))
    maze.append(s)

visited = [[0]*c for _ in range(r)]

def scope(x, y):
    return 0<= x <r and 0<= y <c

@lru_cache(maxsize=None)
```

基本信息
#: 47790196
题目: 01088
提交人: 24n2400011498
内存: 16612kB
时间: 52ms
语言: Python3
提交时间: 2024-12-17 16:34:23

## 25572: 螃蟹采蘑菇

bfs, dfs, http://cs101.openjudge.cn/practice/25572/

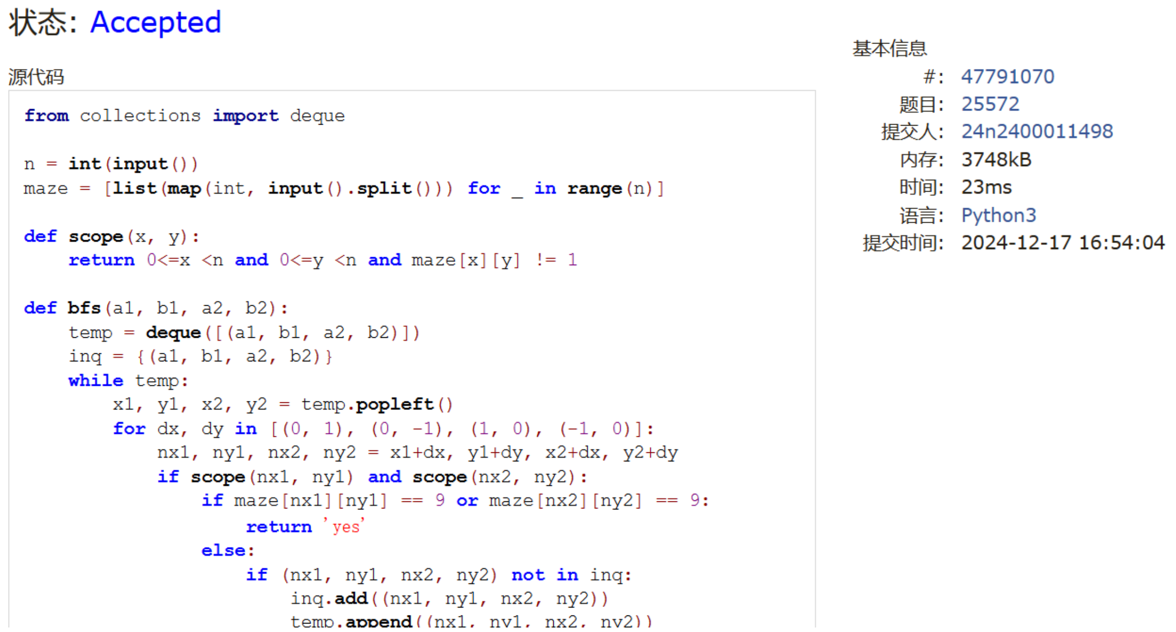
思路:

看出是bfs的变形，把一个坐标变成两个坐标即可。

代码:

```python
from collections import deque

n = int(input())
maze = [list(map(int, input().split())) for _ in range(n)]

def scope(x, y):
    return 0<=x <n and 0<=y <n and maze[x][y] != 1

def bfs(a1, b1, a2, b2):
    temp = deque([(a1, b1, a2, b2)])
    inq = {(a1, b1, a2, b2)}
    while temp:
        x1, y1, x2, y2 = temp.popleft()
        for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
            nx1, ny1, nx2, ny2 = x1+dx, y1+dy, x2+dx, y2+dy
            if scope(nx1, ny1) and scope(nx2, ny2):
                if maze[nx1][ny1] == 9 or maze[nx2][ny2] == 9:
                    return 'yes'
                else:
                    if (nx1, ny1, nx2, ny2) not in inq:
                        inq.add((nx1, ny1, nx2, ny2))
                        temp.append((nx1, ny1, nx2, ny2))
    return 'no'

for i in range(n):
    for j in range(n):
        if maze[i][j] == 5:
            x, y = i, j
            for dx, dy in [(0, 1), (0, -1), (-1, 0), (1, 0)]:
                nx = x+dx
                ny = y+dy
                if scope(nx, ny) and maze[nx][ny] == 5:
                    print(bfs(x, y, nx, ny))
                    exit()
```

代码运行截图（至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque

n = int(input())
maze = [list(map(int, input().split())) for _ in range(n)]

def scope(x, y):
    return 0<=x <n and 0<=y <n and maze[x][y] != 1

def bfs(a1, b1, a2, b2):
    temp = deque([(a1, b1, a2, b2)])
    inq = {(a1, b1, a2, b2)}
    while temp:
        x1, y1, x2, y2 = temp.popleft()
        for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
            nx1, ny1, nx2, ny2 = x1+dx, y1+dy, x2+dx, y2+dy
            if scope(nx1, ny1) and scope(nx2, ny2):
                if maze[nx1][ny1] == 9 or maze[nx2][ny2] == 9:
                    return 'yes'
                else:
                    if (nx1, ny1, nx2, ny2) not in inq:
                        inq.add((nx1, ny1, nx2, ny2))
                        temp.append((nx1, ny1, nx2, ny2))
```

基本信息

| | |
|---|---|
| #: | 47791070 |
| 题目: | 25572 |
| 提交人: | 24n2400011498 |
| 内存: | 3748kB |
| 时间: | 23ms |
| 语言: | Python3 |
| 提交时间: | 2024-12-17 16:54:04 |

## 27373: 最大整数

dp, http://cs101.openjudge.cn/practice/27373/

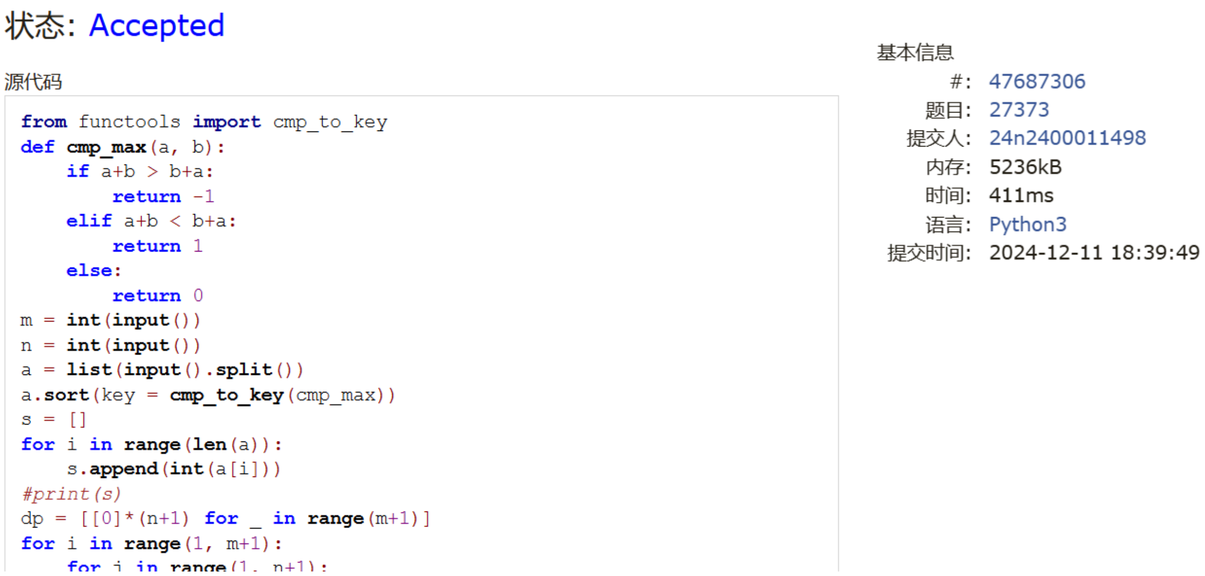
思路：

感觉是最大整数与背包问题的合体，需要同时掌握两件事，感觉不是特别容易想到。

代码：

```python
from functools import cmp_to_key
def cmp_max(a, b):
    if a+b > b+a:
        return -1
    elif a+b < b+a:
        return 1
    else:
        return 0
m = int(input())
n = int(input())
a = list(input().split())
a.sort(key = cmp_to_key(cmp_max))
s = []
for i in range(len(a)):
    s.append(int(a[i]))
#print(s)
dp = [[0]*(n+1) for _ in range(m+1)]
for i in range(1, m+1):
    for j in range(1, n+1):
        if i>=len(str(s[j-1])):
            dp[i][j] = max(dp[i][j-1], int(str(dp[i-len(str(s[j-1]))][j-1])+str(s[j-1])))
        else:
            dp[i][j] = dp[i][j-1]
print(dp[-1][-1])
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from functools import cmp_to_key
def cmp_max(a, b):
    if a+b > b+a:
        return -1
    elif a+b < b+a:
        return 1
    else:
        return 0
m = int(input())
n = int(input())
a = list(input().split())
a.sort(key = cmp_to_key(cmp_max))
s = []
for i in range(len(a)):
    s.append(int(a[i]))
#print(s)
dp = [[0]*(n+1) for _ in range(m+1)]
for i in range(1, m+1):
    for j in range(1, n+1):
```

基本信息

| | |
|---|---|
| #: | 47687306 |
| 题目: | 27373 |
| 提交人: | 24n2400011498 |
| 内存: | 5236kB |
| 时间: | 411ms |
| 语言: | Python3 |
| 提交时间: | 2024-12-11 18:39:49 |

## 02811: 熄灯问题

brute force, http://cs101.openjudge.cn/practice/02811

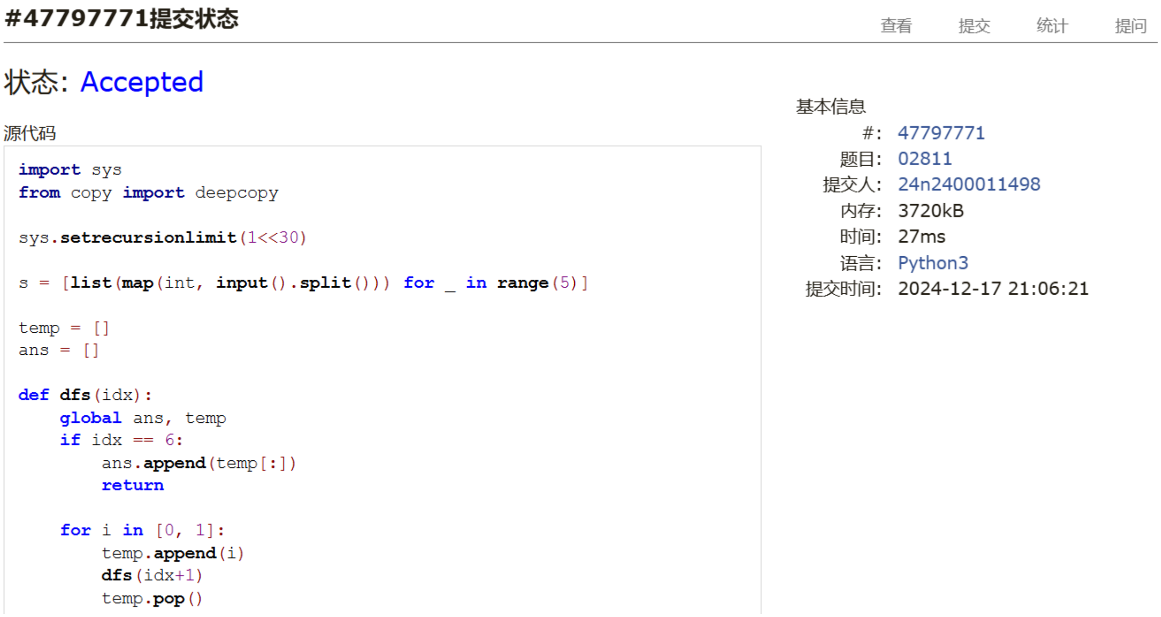
思路：

开始时直接暴力枚举所有可能的情况（对30个点），显然会超时，后来受到题目的启发，发现首先开关按动的顺序与最终结果无关，可以先灭掉第一行在灭掉第二行，所以只需要讨论第一行的操作即可，只有64种情况，剪枝的效果非常好。

代码：

```python
import sys
from copy import deepcopy

sys.setrecursionlimit(1<<30)

s = [list(map(int, input().split())) for _ in range(5)]

temp = []
ans = []

def dfs(idx):
    global ans, temp
    if idx == 6:
        ans.append(temp[:])
        return
    for i in [0, 1]:
        temp.append(i)
        dfs(idx+1)
        temp.pop()
dfs(0)

def scope(x, y):
    return 0<=x<5 and 0<=y<6

for actions in ans:
    test = deepcopy(s)
    for i in range(len(actions)):
        if actions[i] == 1:
            for dx, dy in [(0, 0), (0, 1), (0, -1), (1, 0), (-1, 0)]:
                nx, ny = dx, i+dy
                if scope(nx, ny):
                    test[nx][ny] = (test[nx][ny]+1)%2

    follow_actions = []
    for i in range(1, 5):
        temp = []
        for j in range(6):
            if test[i-1][j] == 1:
                temp.append(1)
                for dx, dy in [(0, 0), (0, 1), (0, -1), (1, 0), (-1, 0)]:
                    nx, ny = i+dx, j+dy
                    if scope(nx, ny):
                        test[nx][ny] = (test[nx][ny] + 1) % 2
            else:
                temp.append(0)
        follow_actions.append(temp[:])

    if sum(test[-1]) == 0:
        print(*actions, sep = ' ')
        for i in range(4):
            print(*follow_actions[i], sep = ' ')
        break
```

代码运行截图 （至少包含有"Accepted"）



#47797771提交状态

状态: Accepted

源代码

```
import sys
from copy import deepcopy

sys.setrecursionlimit(1<<30)

s = [list(map(int, input().split())) for _ in range(5)]

temp = []
ans = []

def dfs(idx):
    global ans, temp
    if idx == 6:
        ans.append(temp[:])
        return

    for i in [0, 1]:
        temp.append(i)
        dfs(idx+1)
        temp.pop()
```

基本信息

| | |
|---|---|
| #: | 47797771 |
| 题目: | 02811 |
| 提交人: | 24n2400011498 |
| 内存: | 3720kB |
| 时间: | 27ms |
| 语言: | Python3 |
| 提交时间: | 2024-12-17 21:06:21 |

## 08210: 河中跳房子

binary search, greedy, http://cs101.openjudge.cn/practice/08210/

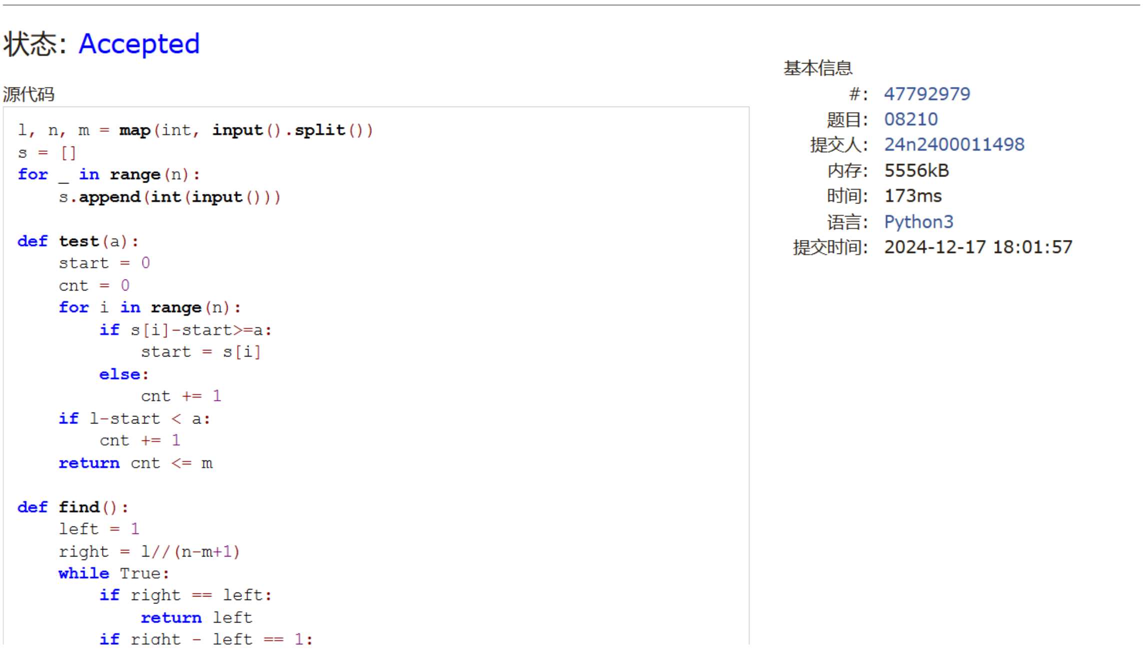
思路：

之前做过aggressive cows, 所以思路还不算困难，但是开始没有注意到l,n,m导致一直出错（难道正常设置变量名不是lmn字母表顺序吗...）

代码：

```
l, n, m = map(int, input().split())
s = []
for _ in range(n):
    s.append(int(input()))

def test(a):
    start = 0
    cnt = 0
    for i in range(n):
        if s[i]-start>=a:
            start = s[i]
        else:
            cnt += 1
    if l-start < a:
        cnt += 1
    return cnt <= m

def find():
    left = 1
    right = l//(n-m+1)
    while True:
        if right == left:
            return left
        if right - left == 1:
            if test(right):
                return right
            else:
                return left
        mid = (left + right) // 2
        if test(mid):
            left = mid
        else:
            right = mid
print(find())
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
l, n, m = map(int, input().split())
s = []
for _ in range(n):
    s.append(int(input()))

def test(a):
    start = 0
    cnt = 0
    for i in range(n):
        if s[i]-start>=a:
            start = s[i]
        else:
            cnt += 1
    if l-start < a:
        cnt += 1
    return cnt <= m

def find():
    left = 1
    right = l//(n-m+1)
    while True:
        if right == left:
            return left
        if right - left == 1:
```

基本信息

#: 47792979
题目: 08210
提交人: 24n2400011498
内存: 5556kB
时间: 173ms
语言: Python3
提交时间: 2024-12-17 18:01:57

## 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。

感觉模板题目掌握的还不错，多数时候甚至还是可以一遍过的，但是没有见过的问题（如brute force剪枝）还是做的非常困难，需要花很多时间才能搞定，一方面还要做些练习，另一方面只能期望上机少些这种题目了...