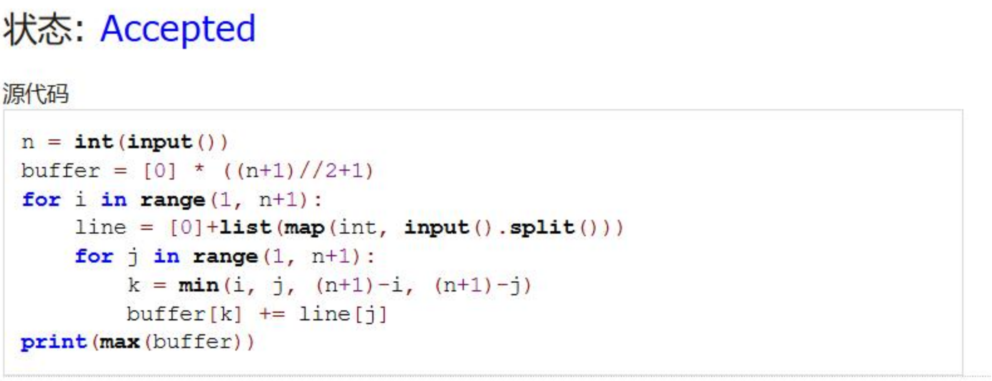
1

2

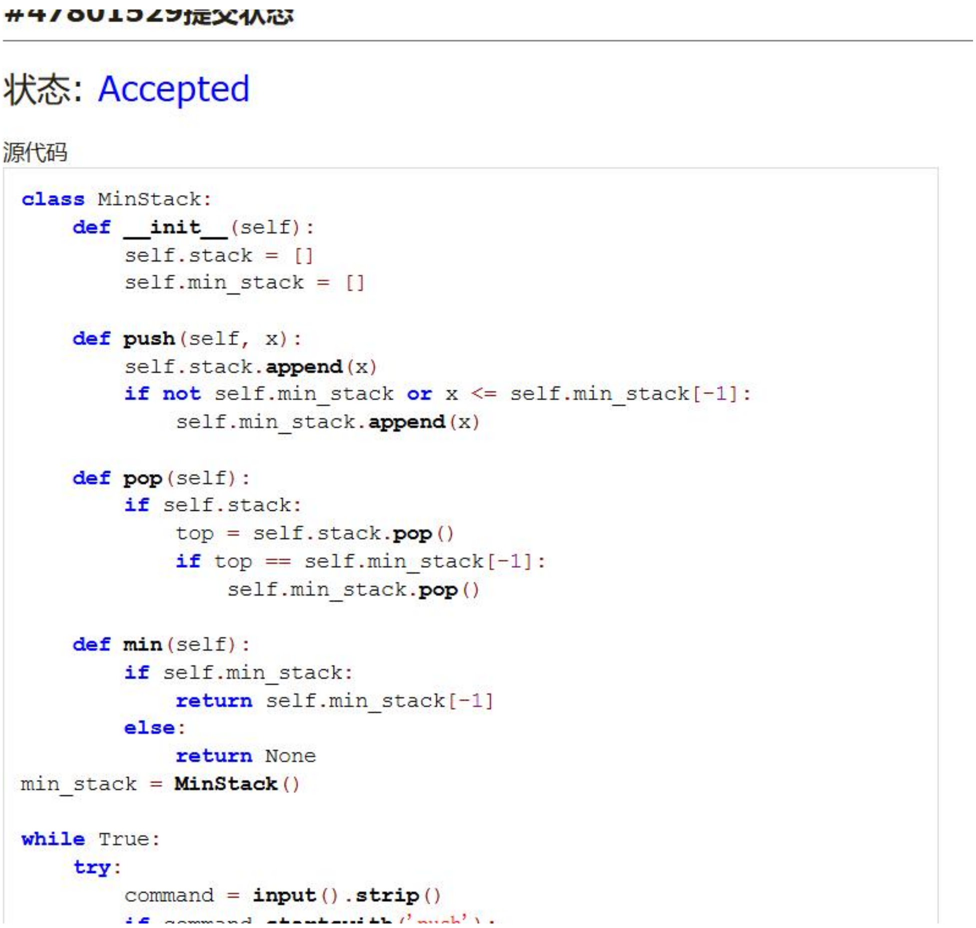


状态: Accepted

源代码

```
n = int(input())
buffer = [0] * ((n+1)//2+1)
for i in range(1, n+1):
    line = [0]+list(map(int, input().split()))
    for j in range(1, n+1):
        k = min(i, j, (n+1)-i, (n+1)-j)
        buffer[k] += line[j]
print(max(buffer))
```

3



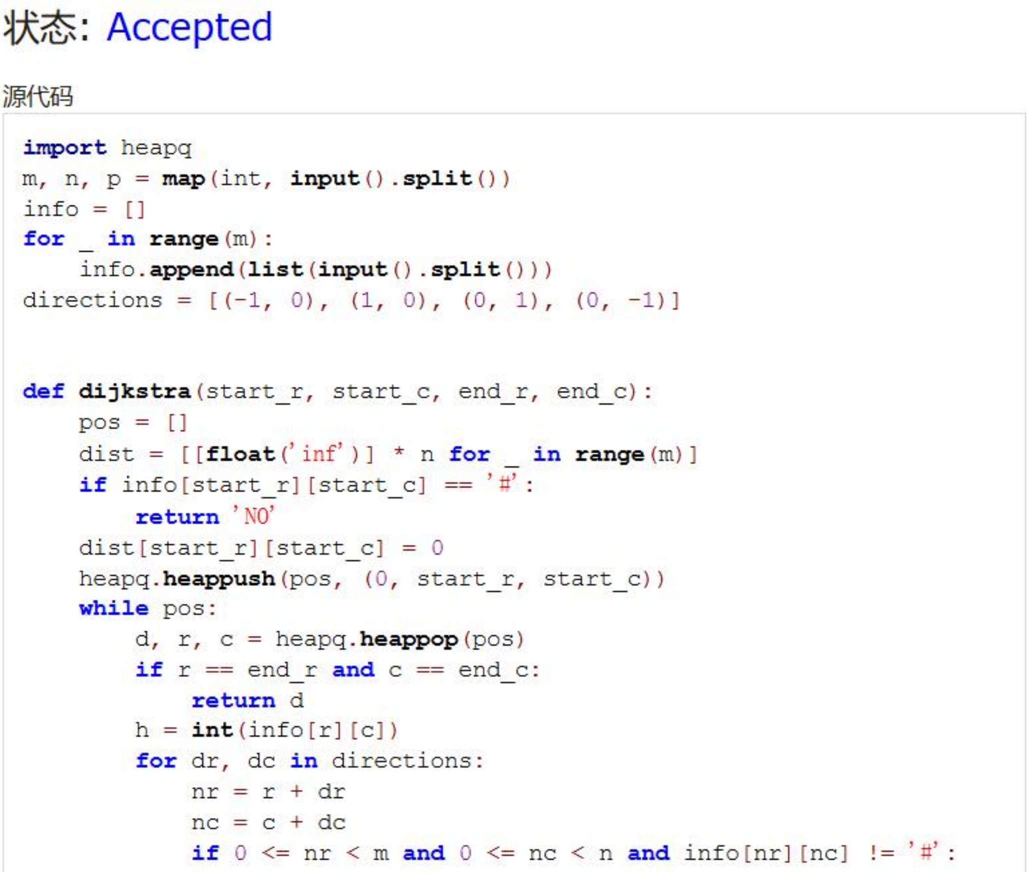
#47801529提交状态

状态: Accepted

源代码

```
class MinStack:
    def __init__(self):
        self.stack = []
        self.min_stack = []

    def push(self, x):
        self.stack.append(x)
        if not self.min_stack or x <= self.min_stack[-1]:
            self.min_stack.append(x)

    def pop(self):
        if self.stack:
            top = self.stack.pop()
            if top == self.min_stack[-1]:
                self.min_stack.pop()

    def min(self):
        if self.min_stack:
            return self.min_stack[-1]
        else:
            return None
min_stack = MinStack()

while True:
    try:
        command = input().strip()
        if command.startswith('push'):
```

4



状态: Accepted

源代码

```
import heapq
m, n, p = map(int, input().split())
info = []
for _ in range(m):
    info.append(list(input().split()))
directions = [(-1, 0), (1, 0), (0, 1), (0, -1)]


def dijkstra(start_r, start_c, end_r, end_c):
    pos = []
    dist = [[float(’inf’)] * n for _ in range(m)]
    if info[start_r][start_c] == ’#’:
        return ’NO’
    dist[start_r][start_c] = 0
    heapq.heappush(pos, (0, start_r, start_c))
    while pos:
        d, r, c = heapq.heappop(pos)
        if r == end_r and c == end_c:
            return d
        h = int(info[r][c])
        for dr, dc in directions:
            nr = r + dr
            nc = c + dc
            if 0 <= nr < m and 0 <= nc < n and info[nr][nc] != ’#’:
```

5

感觉现在做编程有点缺少耐心，还是要在考试之前每天都做几道题，争取多 ACCEPT 几个