1. 题目

E22548: 机智的股民老张

http://cs101.openjudge.cn/practice/22548/

代码：

a = map(int, input().split())  
min\_price = float('inf')  
max\_profit = 0  
  
for price in a:  
 min\_price = min(min\_price, price)

max\_profit = max(max\_profit, price - min\_price)   
  
print(max\_profit)

代码运行截图 （至少包含有"Accepted"）



M28701: 炸鸡排

greedy, http://cs101.openjudge.cn/practice/28701/

代码：

n,k = map(int,input().split())

t = list(map(int,input().split()))

t.sort()

s = sum(t)while True:

if t[-1] > s/k:

s -= t[-1]

t.pop()

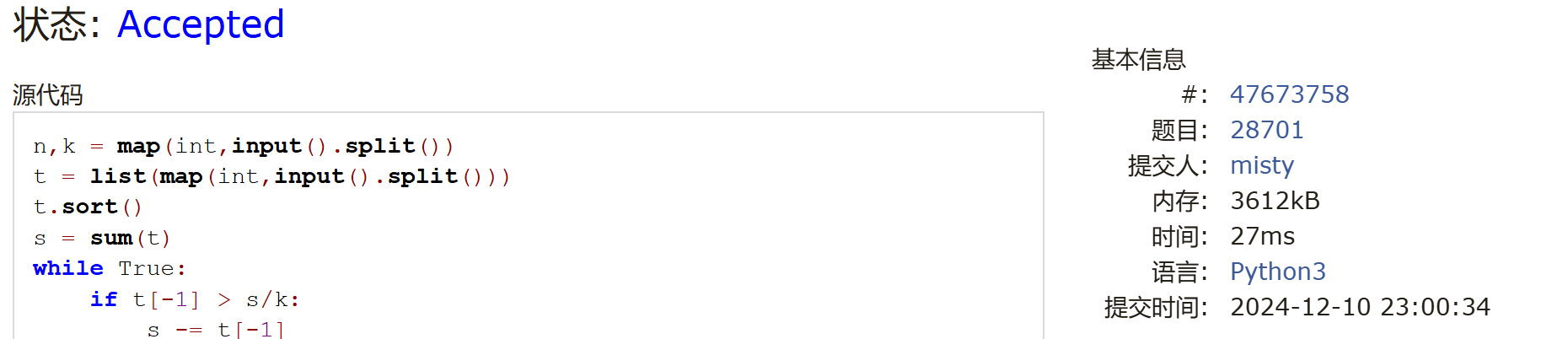
k -= 1

else:

print(f'{s / k:.3f}')

break

代码运行截图 ==（至少包含有"Accepted"）==



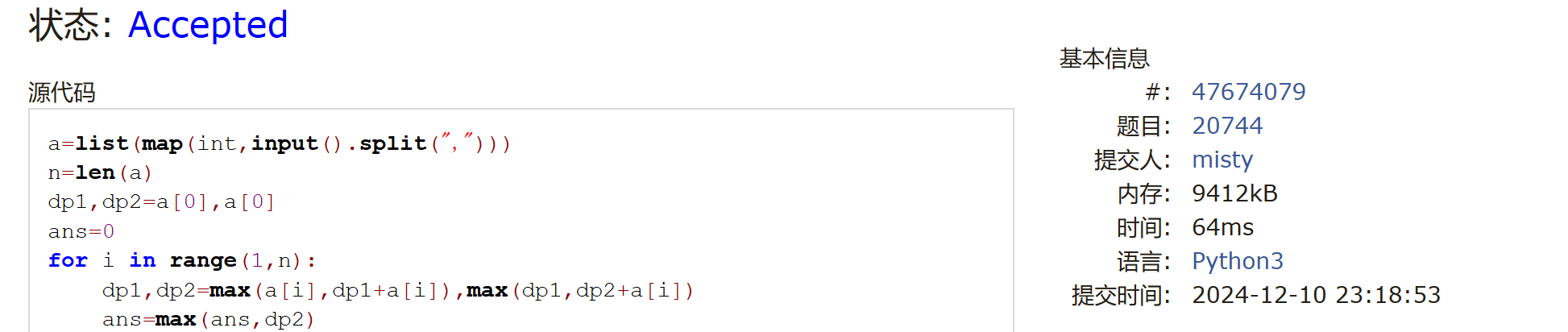
M20744: 土豪购物

dp, http://cs101.openjudge.cn/practice/20744/

代码：

a=list(map(int,input().split(",")))  
n=len(a)  
dp1,dp2=a[0],a[0]  
ans=0  
for i in range(1,n):  
 dp1,dp2=max(a[i],dp1+a[i]),max(dp1,dp2+a[i])  
 ans=max(ans,dp2)  
print(ans)

代码运行截图 （至少包含有"Accepted"）



T25561: 2022决战双十一

brute force, dfs, http://cs101.openjudge.cn/practice/25561/

代码：

result = float("inf")

n, m = map(int, input().split())

store\_prices = [input().split() for \_ in range(n)]

you= [input().split() for \_ in range(m)]

la=[0]\*m

def dfs(i,sum1):

global result

if i==n:

jian=0

for i2 in range(m):

store\_j=0

for k in you[i2]:

a,b=map(int,k.split('-'))

if la[i2]>=a:

store\_j=max(store\_j,b)

jian+=store\_j

result=min(result,sum1-(sum1//300)\*50-jian)

return

for i1 in store\_prices[i]:

idx,p=map(int,i1.split(':'))

la[idx-1]+=p

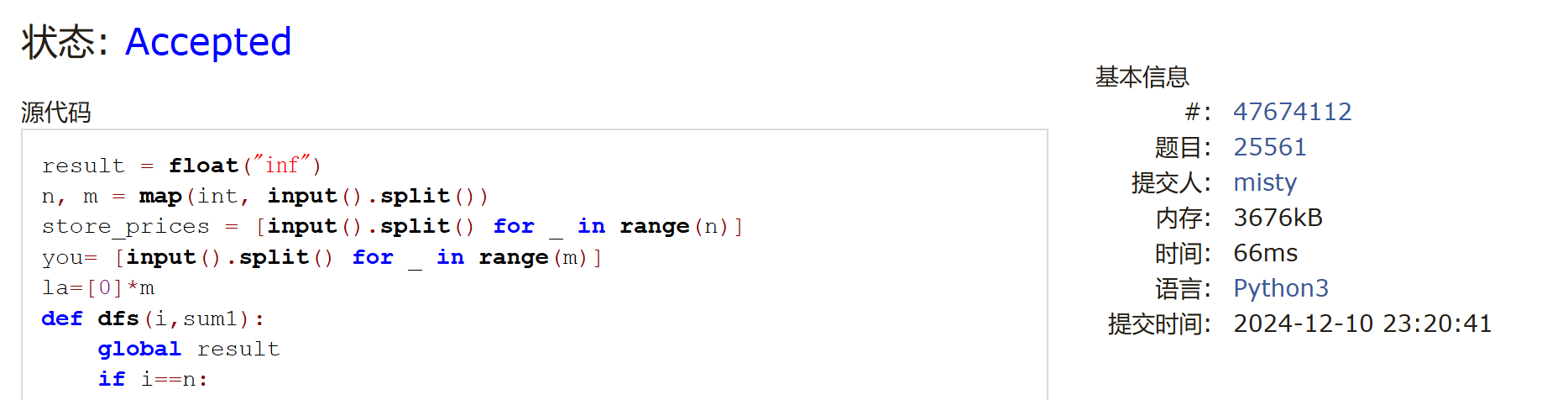
dfs(i+1,sum1+p)

la[idx-1]-=p

dfs(0,0)

print(result)

代码运行截图 （至少包含有"Accepted"）



T20741: 两座孤岛最短距离

dfs, bfs, http://cs101.openjudge.cn/practice/20741/

代码：

from collections import deque

def dfs(x, y, grid, n, queue, directions):

""" Mark the connected component starting from (x, y) as visited using DFS. """

grid[x][y] = 2 # Mark as visited

queue.append((x, y))

for dx, dy in directions:

nx, ny = x + dx, y + dy

if 0 <= nx < n and 0 <= ny < n and grid[nx][ny] == 1:

dfs(nx, ny, grid, n, queue, directions)

def bfs(grid, n, queue, directions):

""" Perform BFS to find the shortest path to another component. """

distance = 0

while queue:

for \_ in range(len(queue)):

x, y = queue.popleft()

for dx, dy in directions:

nx, ny = x + dx, y + dy

if 0 <= nx < n and 0 <= ny < n:

if grid[nx][ny] == 1:

return distance

elif grid[nx][ny] == 0:

grid[nx][ny] = 2 # Mark as visited

queue.append((nx, ny))

distance += 1

return distance

def main():

n = int(input())

grid = [list(map(int, input())) for \_ in range(n)]

directions = [(1, 0), (-1, 0), (0, 1), (0, -1)]

queue = deque()

# Start DFS from the first '1' found and use BFS from there

for i in range(n):

for j in range(n):

if grid[i][j] == 1:

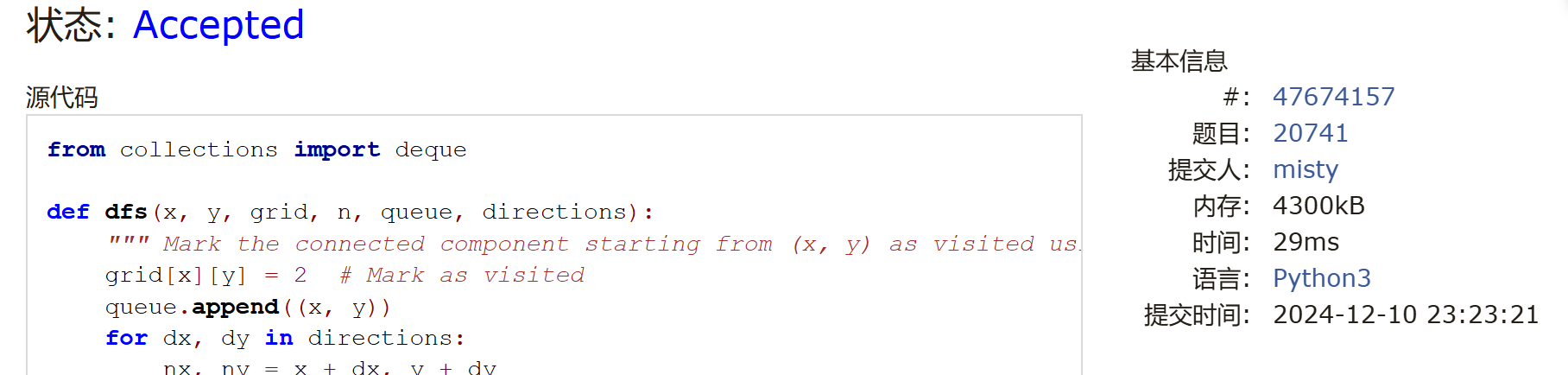
dfs(i, j, grid, n, queue, directions)

return bfs(grid, n, queue, directions)

if \_\_name\_\_ == "\_\_main\_\_":

print(main())

代码运行截图 （至少包含有"Accepted"）



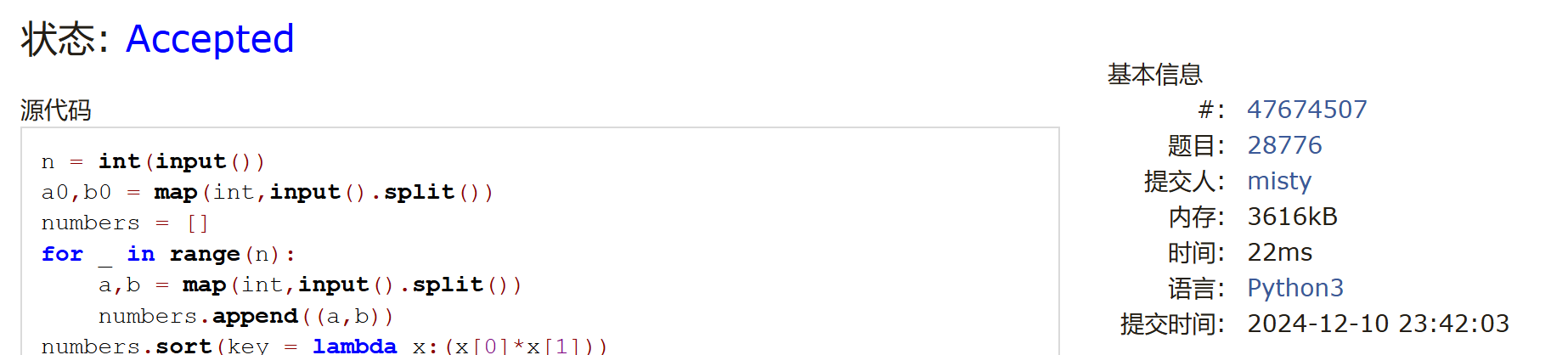
T28776: 国王游戏

greedy, http://cs101.openjudge.cn/practice/28776

代码：

n = int(input())  
a0,b0 = map(int,input().split())  
numbers = []  
for \_ in range(n):  
 a,b = map(int,input().split())  
 numbers.append((a,b))  
numbers.sort(key = lambda x:(x[0]\*x[1]))  
result = 0  
for i in range(n):  
 result = max(result,a0 //numbers[i][1])  
 a0 \*= numbers[i][0]  
print(result)

代码运行截图 （至少包含有"Accepted"）



1. 学习总结和收获

1现在比较会的就dp了

2 bfsdfs一长就感觉难学

3 贪心写起来很公式化，就是不好想排序的条件，比如国王游戏