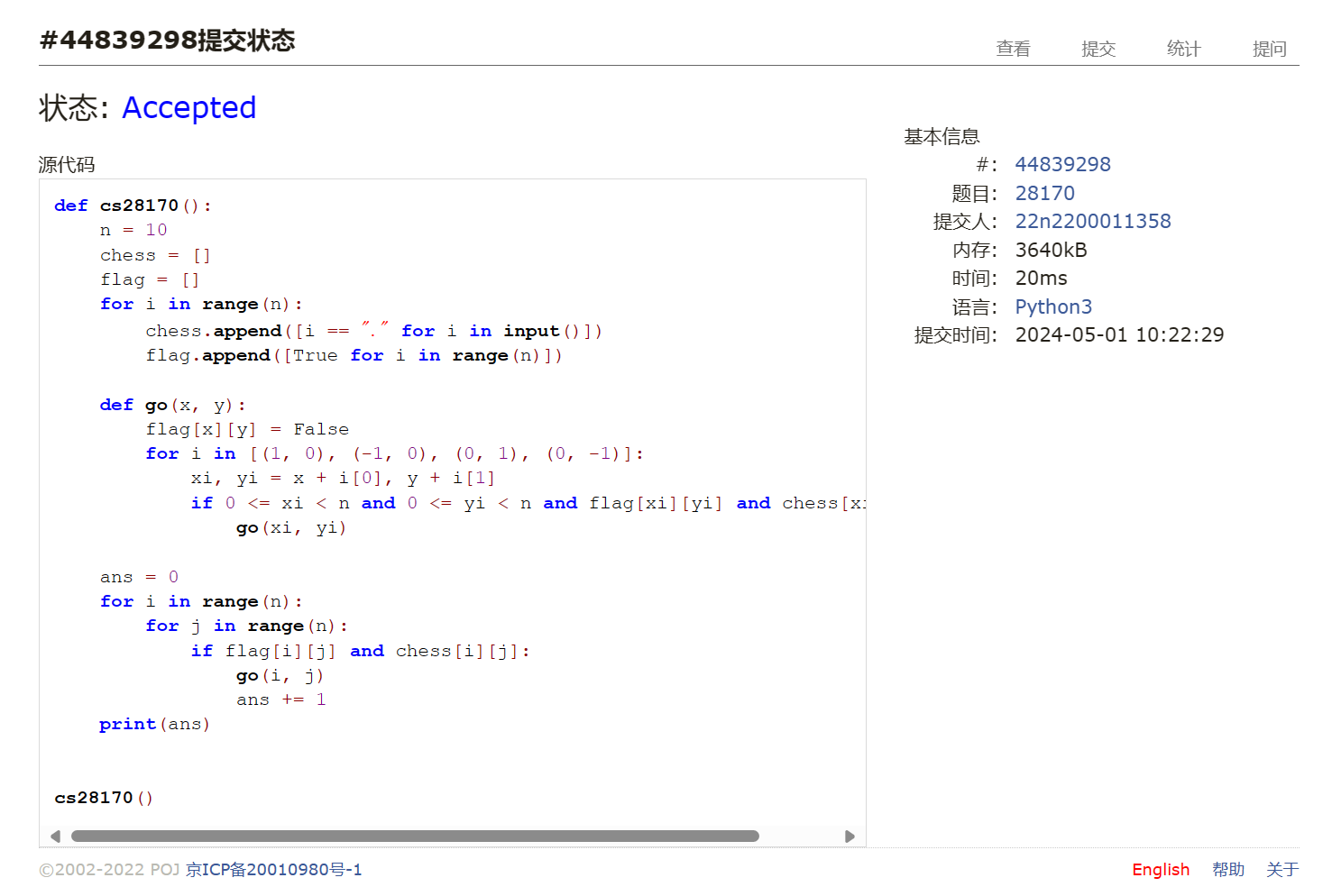
**28170:算鹰**

def cs28170():  
 n = 10  
 chess = []  
 flag = []  
 for i in range(n):  
 chess.append([i == "." for i in input()])  
 flag.append([True for i in range(n)])  
  
 def go(x, y):  
 flag[x][y] = False  
 for i in [(1, 0), (-1, 0), (0, 1), (0, -1)]:  
 xi, yi = x + i[0], y + i[1]  
 if 0 <= xi < n and 0 <= yi < n and flag[xi][yi] and chess[xi][yi]:  
 go(xi, yi)  
  
 ans = 0  
 for i in range(n):  
 for j in range(n):  
 if flag[i][j] and chess[i][j]:  
 go(i, j)  
 ans += 1  
 print(ans)



## 18250:冰阔落 I

def cs18250():  
 try:  
 while True:  
 n, m = map(int, input().split())  
 bottle = [[i] for i in range(n)]  
 cola = [i for i in range(n)]  
 for i in range(m):  
 x, y = map(lambda j: int(j) - 1, input().split())  
 if cola[x] == cola[y]:  
 print("Yes")  
 else:  
 print("No")  
 tmp = cola[y]  
 while bottle[tmp]:  
 pour = bottle[tmp].pop()  
 bottle[cola[x]].append(pour)  
 cola[pour] = cola[x]  
 print(bottle, cola)  
 ans = []  
 for i in range(n):  
 if bottle[i]:  
 ans.append(i)  
 print(len(ans))  
 for i in ans[:-1]:  
 print(i + 1, end=" ")  
 print(ans[-1] + 1)  
 except:  
 pass

## 05907:二叉树的操作

def cs05907():  
 for t in range(int(input())):  
 n, m = map(int, input().split())  
 dct = {}  
 f = {}  
 for i in range(n):  
 x, y, z = map(int, input().split())  
 dct[x] = [y, z]  
 f[y] = [x, 0]  
 f[z] = [x, 1]  
 for i in range(m):  
 ipt = tuple(map(int, input().split()))  
 if ipt[0] == 1:  
 dct[f[ipt[1]][0]][f[ipt[1]][1]] = ipt[2]  
 dct[f[ipt[2]][0]][f[ipt[2]][1]] = ipt[1]  
 f[ipt[1]], f[ipt[2]] = f[ipt[2]], f[ipt[1]]  
 else:  
 x = ipt[1]  
 while dct[x][0] + 1:  
 x = dct[x][0]  
 print(x)



## 03151:Pots

def cs03151():  
 a, b, c = map(int, input().split())  
 memory = set()  
 queue = [[(0, 0), []]]  
 while queue:  
 nxt = []  
 while queue:  
 i = queue.pop()  
 if i[0][0] == c or i[0][1] == c:  
 print(len(i[1]))  
 for j in i[1]:  
 print(j)  
 exit()  
 if i[0] not in memory:  
 memory.add(i[0])  
 if i[0][0]:  
 nxt.append([(0, i[0][1]), i[1] + ["DROP(1)"]])  
 nxt.append([(max(0, i[0][0] + i[0][1] - b), min(i[0][0] + i[0][1], b)), i[1] + ["POUR(1,2)"]])  
 if i[0][1]:  
 nxt.append([(i[0][0], 0), i[1] + ["DROP(2)"]])  
 nxt.append([(min(i[0][0] + i[0][1], a), max(0, i[0][0] + i[0][1] - a)), i[1] + ["POUR(2,1)"]])  
 if i[0][0] - a:  
 nxt.append([(a, i[0][1]), i[1] + ["FILL(1)"]])  
 if i[0][1] - b:  
 nxt.append([(i[0][0], b), i[1] + ["FILL(2)"]])  
  
 queue = nxt  
 print("impossible")



## 05443:兔子与樱花

def cs05442():  
 dct = {}  
 n = int(input()) - 1  
 for i in range(n):  
 ipt = input().split()  
 dct[ipt[0]] = []  
 for j in range(int(ipt[1])):  
 dct[ipt[0]].append((ipt[j \* 2 + 2], int(ipt[j \* 2 + 3])))  
 s = "ABCDEFGHIJKLMNOPQRSTUVWXYZa"  
 tmp = []  
 for i in range(n):  
 tmp += [(i, s.find(j[0]), j[1]) for j in dct[s[i]]]  
 tmp.sort(key=lambda i: i[2])  
 dest = [-1 for i in range(n + 1)]  
 ans, j = 0, 0  
 for i in tmp:  
 iid, jid = i[0], i[1]  
 while dest[iid] + 1:  
 iid = dest[iid]  
 while dest[jid] + 1:  
 jid = dest[jid]  
 if iid - jid:  
 dest[iid] = jid  
 ans += i[2]  
 j += 1  
 if j == n:  
 break  
  
 print(ans)

## 02754:八皇后

def cs02754():  
 c = "12345678"  
 a = []  
  
 def go(ans, n):  
 if n == 8:  
 a.append(ans)  
 for i in c:  
 if ans.find(i) == -1:  
 k, flag = 0, True  
 for j in ans:  
 if int(i) - n == int(j) - k or int(i) + n == int(j) + k:  
 flag = False  
 break  
 k += 1  
 if flag:  
 go(ans + i, n + 1)  
  
 go("", 0)  
 for i in range(int(input())):  
 print(a[int(input()) - 1])  
  
  
cs02754()



总结：本次作业难度低于上次，想明白以后基本一遍过，POT那题多花了点时间，是看到bfs的提示并且数据量比较小才做出来的