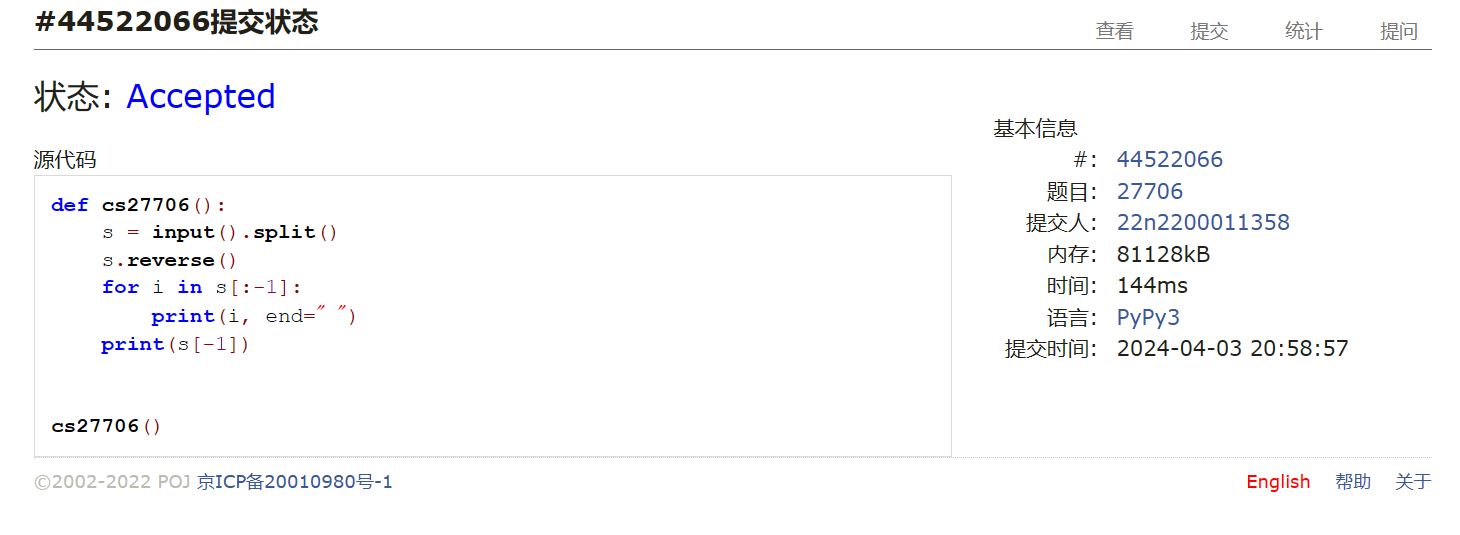
**27706:逐词倒放**

def cs27706():  
 s = input().split()  
 s.reverse()  
 for i in s[:-1]:  
 print(i, end=" ")  
 print(s[-1])



## 27951:机器翻译

def cs27951():  
 m, n = map(int, input().split())  
 ipt = list(map(int, input().split()))  
 lst = []  
 ans = 0  
 for i in ipt:  
 if i in lst:  
 pass  
 else:  
 lst.append(i)  
 ans += 1  
 if len(lst) > m:  
 del lst[0]  
 print(ans)



## 27932:Less or Equal

def cs27932():  
 n, k = map(int, input().split())  
 a = list(map(int, input().split()))  
 a.sort()  
 if k == 0:  
 if a[0] == 1:  
 print(-1)  
 else:  
 print(1)  
 elif n == k:  
 print(a[k - 1])  
 elif a[k - 1] == a[k]:  
 print(-1)  
 elif a[k - 1] > 10 \*\* 9:  
 print(-1)  
 else:  
 print(a[k - 1])



## 27948:FBI树

def cs27948():  
 n = int(input())  
 ans = [""]  
  
 def fbi(s):  
 if "1" not in s:  
 ans[0] = "B" + ans[0]  
 elif "0" not in s:  
 ans[0] = "I" + ans[0]  
 else:  
 ans[0] = "F" + ans[0]  
 l = len(s)  
 if l >= 2:  
 fbi(s[l // 2:])  
 fbi(s[:l // 2])  
  
 fbi(input())  
 print(ans[0])



## 27928:遍历树

def cs27928():  
 n = int(input())  
 dct = {}  
 key, leef = set(), set()  
 for i in range(n):  
 lst = list(map(int, input().split()))  
 k = lst[0]  
 key.add(k)  
 if len(lst) > 1:  
 for i in lst[1:]:  
 leef.add(i)  
 lst.sort()  
 dct[k] = lst  
 root = list(key - leef)[0]  
 ans = []  
  
 def goto(ky):  
 for j in dct[ky]:  
 ans.append(j)  
 if j != ky:  
 goto(j)  
  
 goto(root)  
 for i in leef:  
 ans.remove(i)  
 for i in ans:  
 print(i)



## 27925:小组队列

def cs27925():  
 t = int(input())  
 team, tf = [], []  
 lst = []  
 for i in range(t):  
 team.append(input().split())  
 while True:  
 ipt = input()  
 if ipt[0] == "S":  
 break  
 elif ipt[0] == "E":  
 ipt = ipt.split()[1]  
 ti = 0  
 for i in team:  
 if ipt in i:  
 break  
 ti += 1  
 if ti not in tf:  
 tf.append(ti)  
 lst.append([ipt])  
 else:  
 lst[tf.index(ti)].append(ipt)  
 elif ipt[0] == "D":  
 print(lst[0].pop(0))  
 if not lst[0]:  
 lst.pop(0)  
 tf.pop(0)  
  
  
cs27925()



总结：本次两个T题是对时间复杂度和递归的简单考察。月考题不难，用心做都能AC,2小时内能AC5，AC第六题时超时了20分钟，仍需加油。