1. 题目

sy119: 汉诺塔

recursion, <https://sunnywhy.com/sfbj/4/3/119>

代码：

def moveHanoi(n, from\_rod, to\_rod, mid\_rod):

    if n == 0:

        return

    moveHanoi(n - 1, from\_rod, mid\_rod, to\_rod)

    print(f"{from\_rod}->{to\_rod}")

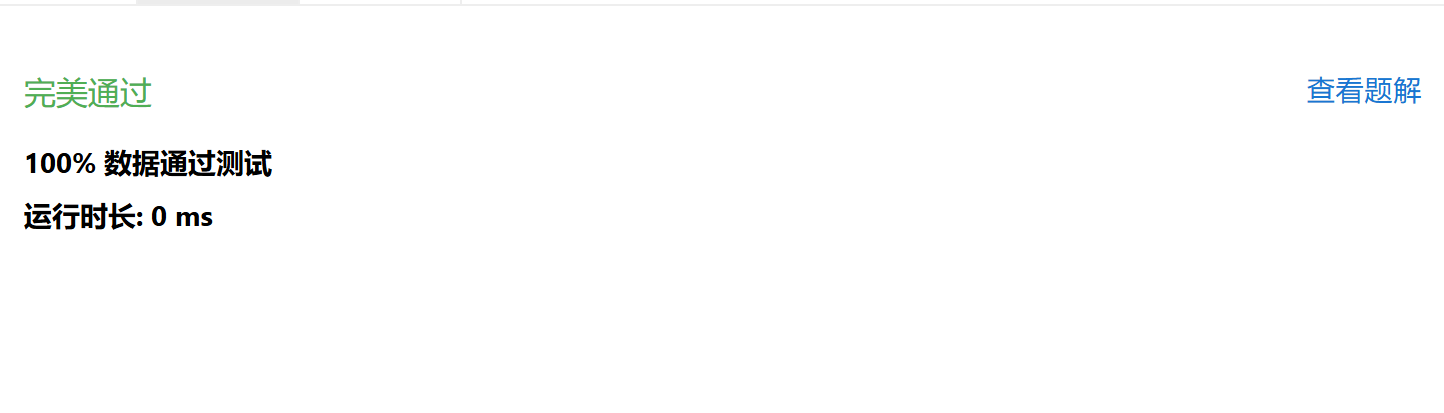
    moveHanoi(n - 1, mid\_rod, to\_rod, from\_rod)

n = int(input())

print(2\*\*n - 1)

moveHanoi(n, 'A', 'C', 'B')

代码运行截图 （至少包含有"Accepted"）



sy132: 全排列I

recursion, <https://sunnywhy.com/sfbj/4/3/132>

代码：

def dfs(idx, n, used, temp, result):

    if idx == n + 1:

        result.append(temp[:])

        return

    for i in range(1, n + 1):

        if not used[i]:

            temp.append(i)

            used[i] = True

            dfs(idx + 1, n, used, temp, result)

            used[i] = False

            temp.pop()

def generate\_permutations(n):

    result = []

    used = [False] \* (n + 1)

    dfs(1, n, used, [], result)

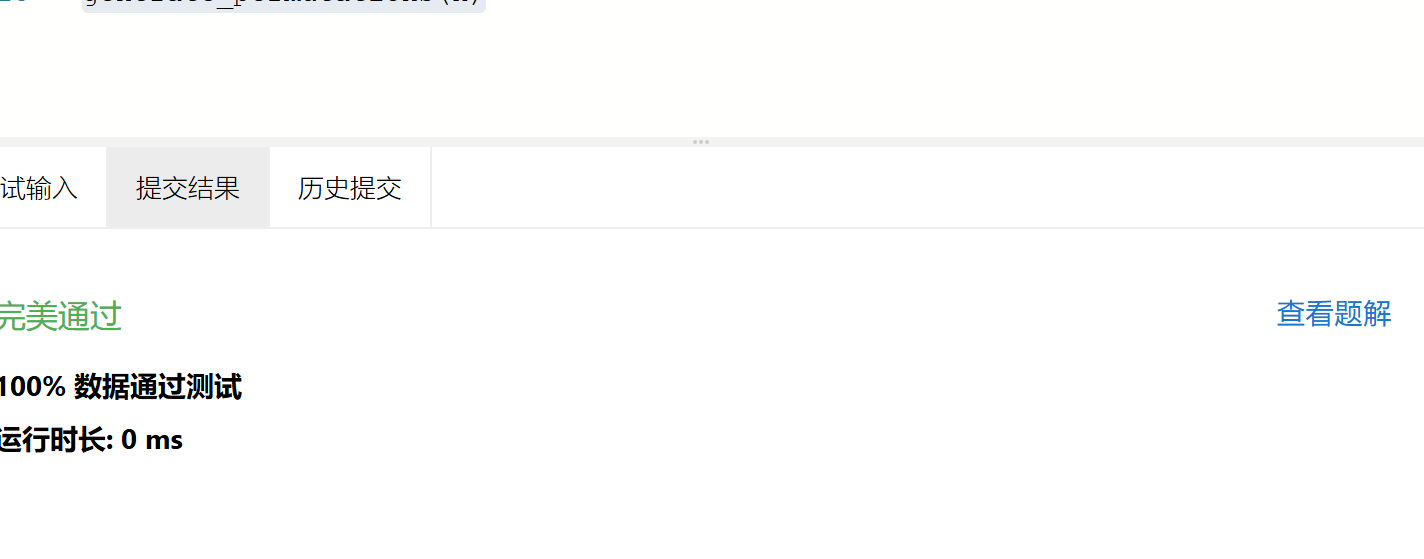
    for perm in result:

        print(" ".join(map(str, perm)))

n = int(input())

generate\_permutations(n)

代码运行截图 ==（至少包含有"Accepted"）==



02945: 拦截导弹

dp, <http://cs101.openjudge.cn/2024fallroutine/02945>

代码：

def max\_intercepted\_missiles(k, heights):

# Initialize the dp array

dp = [1] \* k

# Fill the dp array

for i in range(1, k):

for j in range(i):

if heights[i] <= heights[j]:

dp[i] = max(dp[i], dp[j] + 1)

return max(dp)

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

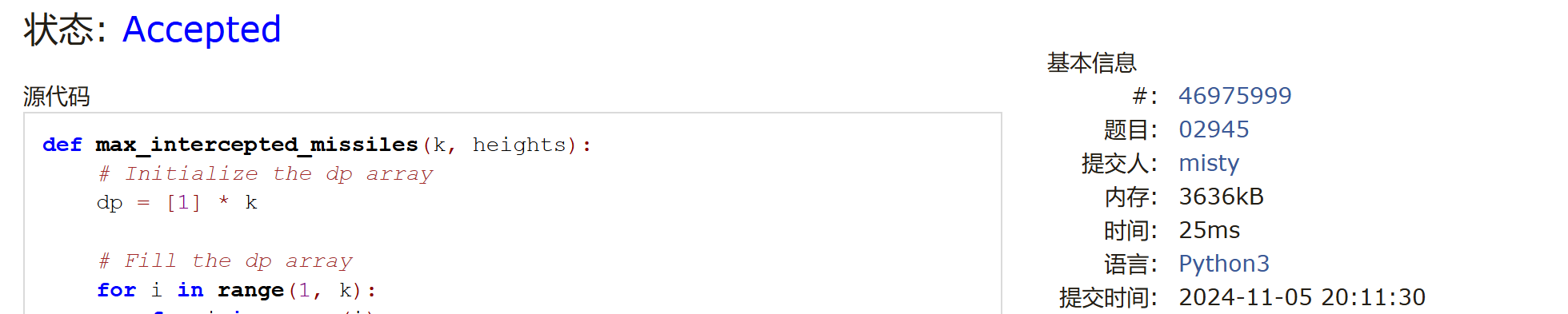
k = int(input())

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

result = max\_intercepted\_missiles(k, heights)

print(result)

代码运行截图 （至少包含有"Accepted"）



23421: 小偷背包

dp, <http://cs101.openjudge.cn/practice/23421>

代码：

N,B = map(int, input().split())\*p, = map(int, input().split())\*w, = map(int, input().split())

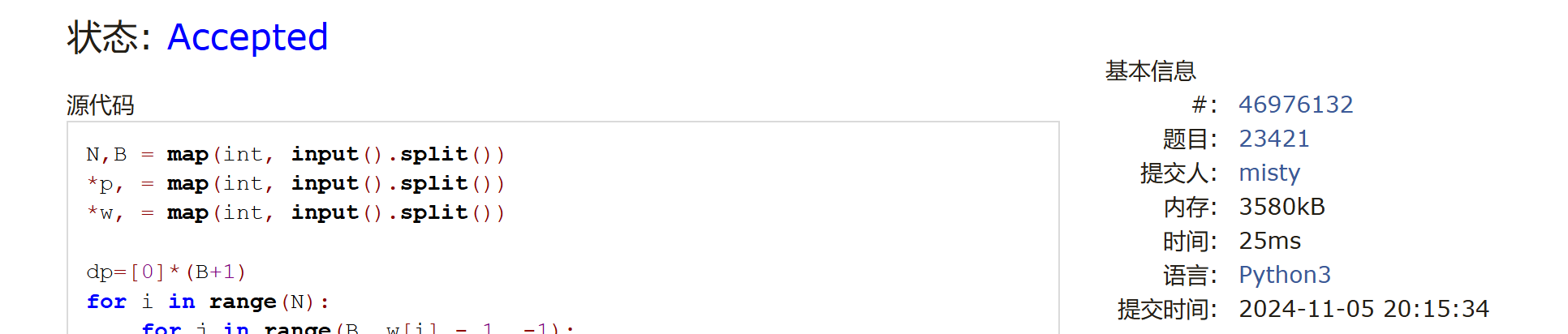
dp=[0]\*(B+1)for i in range(N):

for j in range(B, w[i] - 1, -1):

dp[j] = max(dp[j], dp[j-w[i]]+p[i])

print(dp[-1])

代码运行截图 （至少包含有"Accepted"）



02754: 八皇后

dfs and similar, <http://cs101.openjudge.cn/practice/02754>

代码：

list1 = []

def queen(s):

if len(s) == 8:

list1.append(s)

return

for i in range(1, 9):

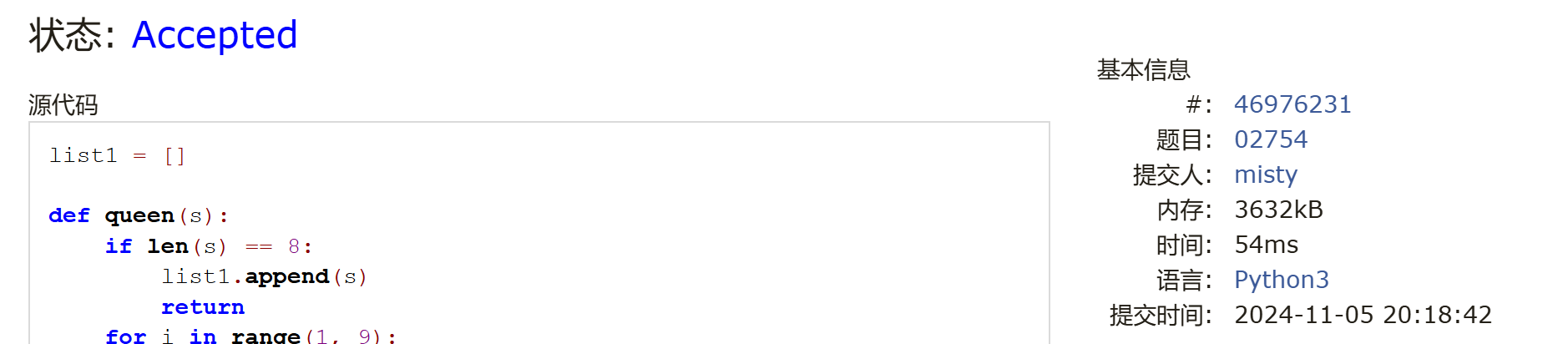
if all(str(i) != s[j] and abs(len(s) - j) != abs(i - int(s[j])) for j in range(len(s))):

queen(s + str(i))

queen('')samples = int(input())for k in range(samples):

print(list1[int(input()) - 1])

代码运行截图 （至少包含有"Accepted"）



189A. Cut Ribbon

brute force, dp 1300 <https://codeforces.com/problemset/problem/189/A>

代码：

n, a, b, c = map(int, input().split())dp = [0]+[float('-inf')]\*n

for i in range(1, n+1):

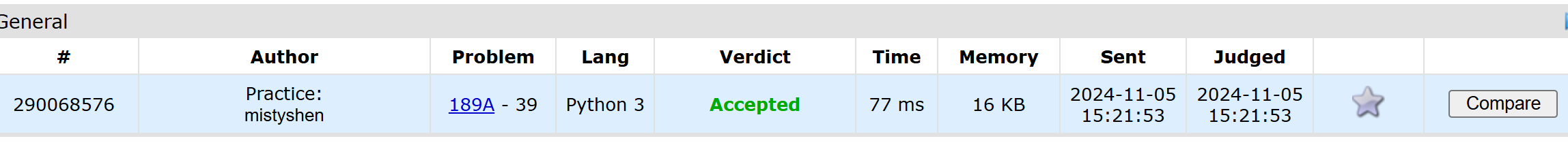
for j in (a, b, c):

if i >= j:

dp[i] = max(dp[i-j] + 1, dp[i])

print(dp[n])

代码运行截图 （至少包含有"Accepted"）



1. 学习总结和收获
2. 期中考完再学DP
3. 往期每日练习还有一部分没有做
4. 看老师上课时的讲义，再结合网上搜索的资料和ai基本可以完成自学，但是对于零基础开始的我来说时间紧是比较大的问题。