

Assignment #B: 贪心、矩阵和动态规划

Updated 0118 GMT+8 Nov 21, 2023

2023 fall, Compiled by Xinjie Song, Phy

说明:

本周作业留点难题，期中考试结束了，需要学习计算概论了。这次不分必做选做题了，如果耗时太长，直接找答案看。两个题解，经常更新。所以最好从这个链接下载最新的，<https://github.com/GMyhf/2020fall-cs101>。

1) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted, 学号），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用 word）。AC 或者没有AC，都请标上每个题目大致花费时间。

2) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、作业评论有md或者doc。

3) 如果不能在截止前提交作业，请写明原因。

编程环境

操作系统: Windows 11 22H2

Python编程环境: PyCharm 2023.2 (Community Edition)

C/C++编程环境: g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

如果耗时太长，直接看解题思路，或者源码

02786:Pell数列

<http://cs101.openjudge.cn/practice/02786/>

思路：找到周期，直接输出

代码

```
ls = [1, 2]
while True:
    new = (2*ls[-1] + ls[-2]) % 32767
    if new == 2 and ls[-1] == 1:
        t = len(ls) - 1
        ls.pop()
        break
    ls.append(new)
for _ in range(int(input())):
    print(ls[(int(input()) - 1) % t])
```

代码运行截图



04133:垃圾炸弹

matrices, <http://cs101.openjudge.cn/practice/04133/>

思路：将某个垃圾能被清除的范围内的矩阵值加一，最后整个矩阵值最大的点便是所求点，统计并输出即可

代码

```
maps = [[None for j in range(1025)] for i in range(1025)]
d, n = int(input()), int(input())
max_num = 0
count = 0
for i in range(n):
    x, y, num = map(int, input().split())
```

```

for j in range(max(x - d, 0), min(x + d + 1, 1025)):
    for k in range(max(y - d, 0), min(y + d + 1, 1025)):
        if maps[j][k]:
            maps[j][k] += num
        else:
            maps[j][k] = num
        if maps[j][k] > max_num:
            max_num = maps[j][k]
            count = 1
        elif maps[j][k] == max_num:
            count += 1
print(count, max_num)

```

代码运行截图



26971:分发糖果

greedy, <http://cs101.openjudge.cn/routine/26971/>

思路：分别正向、反向遍历数据，标记单增和单减部分即可

代码

```

n = int(input())
ratings = [int(i) for i in input().split()]
candies = [0 for i in range(n)]
i = 0
while i < n:
    j = i + 1
    while j < n:
        if ratings[j] <= ratings[j - 1]:

```

```

        break
    candies[j - 1] = j - i
    if j == n - 1:
        candies[j] = j - i + 1
    j += 1
    i = j
i = 0
while i < n:
    j = i + 1
    while j < n:
        if ratings[n - 1 - j] <= ratings[n - j]:
            break
        candies[n - j] = j - i
        if j == n - 1:
            candies[n - 1 - j] = j - i + 1
        j += 1
    i = j
candies[0] = [candies[0], 1][candies[0] == 0]
candies[-1] = [candies[-1], 1][candies[-1] == 0]

for i in range(1, n - 1):
    if candies[i] == 0:
        candies[i] = 1 + [max(candies[i - 1], candies[i + 1]), 0][ratings[i] ==
ratings[i - 1] == ratings[i + 1]]
print(sum(candies))

```

代码运行截图

OpenJudge 提交状态

cs101.openjudge.cn/routine/solution/41642797/

OpenJudge 题目ID, 标题, 描述 23n2300011524 信箱 账号

CS101 / 每日选做

题目 排名 状态 提问

#41642797提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

n = int(input())
ratings = [int(i) for i in input().split()]
candies = [0 for i in range(n)]
i = 0
while i < n:
    j = i + 1
    while j < n:
        if ratings[j] <= ratings[j - 1]:
            break
        candies[j - 1] = j - i
        if j == n - 1:
            candies[j] = j - i + 1
        j += 1
    i = j
i = 0
while i < n:
    j = i + 1
    while j < n:
        if ratings[n - 1 - j] <= ratings[n - j]:
            break
        candies[n - j] = j - i
        if j == n - 1:
            candies[n - 1 - j] = j - i + 1
        j += 1
    i = j
candies[0] = [candies[0], 1][candies[0] == 0]
candies[-1] = [candies[-1], 1][candies[-1] == 0]

for i in range(1, n - 1):
    if candies[i] == 0:
        candies[i] = 1 + [max(candies[i - 1], candies[i + 1]), 0][ratings[i] ==
ratings[i - 1] == ratings[i + 1]]
print(sum(candies))

```

基本信息

#: 41642797
 题目: 26971
 提交人: 23n2300011524
 内存: 3828Kb
 时间: 31ms
 语言: Python3
 提交时间: 2023-10-14 10:34:24

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

26976:摆动序列

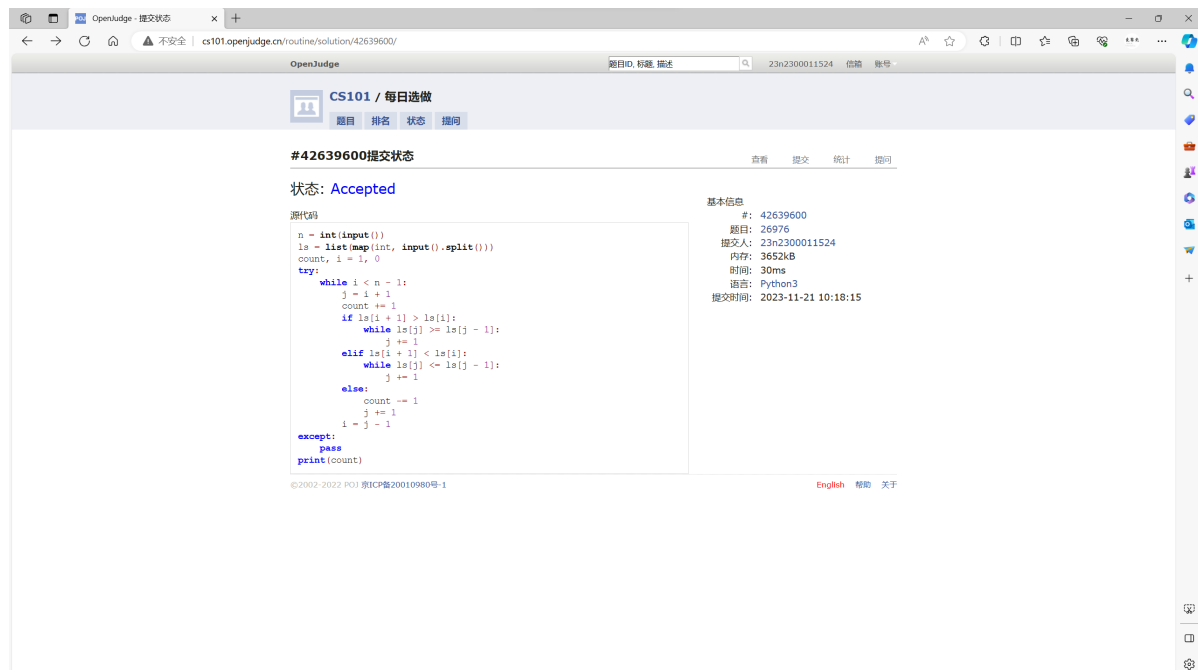
greedy, <http://cs101.openjudge.cn/routine/26976/>

思路：统计单增单减片段数量

代码

```
n = int(input())
ls = list(map(int, input().split()))
count, i = 1, 0
try:
    while i < n - 1:
        j = i + 1
        count += 1
        if ls[i + 1] > ls[i]:
            while ls[j] >= ls[j - 1]:
                j += 1
        elif ls[i + 1] < ls[i]:
            while ls[j] <= ls[j - 1]:
                j += 1
        else:
            count -= 1
            j += 1
        i = j - 1
except:
    pass
print(count)
```

代码运行截图



27104:世界杯只因

<http://cs101.openjudge.cn/practice/27104/>

思路：经hrc同学启发，首先用列表存储每个摄像头的左右端点，根据自定义规则排序：左端点更小的在前，左端点相同的右端点靠后的在前；然后依次遍历排序后的列表，记录右侧可以达到的最大值，当索引超出当前右端时，数量加1，当前右端更改为此时右端点的最大值。

代码

```
n = int(input())
ls = list(map(int, input().split()))
for i in range(n):
    ls[i] = [max(0, i - ls[i]), min(n - 1, i + ls[i])]
ls.sort(key=lambda t: (t[0], -t[1]))
count = 1
left = right = ls[0][1]
for i in range(1, n):
    right = max(right, ls[i][1])
    if ls[i][0] >= left + 1:
        left = right
        count += 1
print(count)
```

代码运行截图

The screenshot shows the OpenJudge submission page for problem 27104. The page is titled "CS101 / 题库" and shows the submission status as "Accepted". The source code is displayed on the left, and the basic information is on the right.

Source Code:

```
n = int(input())
ls = list(map(int, input().split()))
for i in range(n):
    ls[i] = [max(0, i - ls[i]), min(n - 1, i + ls[i])]
ls.sort(key=lambda t: (t[0], -t[1]))
count = 1
left = right = ls[0][1]
for i in range(1, n):
    right = max(right, ls[i][1])
    if ls[i][0] >= left + 1:
        left = right
        count += 1
print(count)
```

Basic Information:

- #: 42647394
- 题目: 27104
- 提交人: 23n2300011524
- 内存: 129868kB
- 时间: 1307ms
- 语言: Python3
- 提交时间: 2023-11-21 16:26:35

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

CF1000B: Light It Up

greedy, 1500, <https://codeforces.com/problemset/problem/1000/B>

思路：统计在每个关灯初后一个位置插入（如果可以插入的话）能够增加的开灯时间，然后找到增量的最大值即可。

代码

```
n, m = map(int, input().split())
ls = [0] + list(map(int, input().split())) + [m]
on = off = 0
add = [0]
if n % 2:
    for i in range(n // 2 + 1):
        on += ls[2*i + 1] - ls[2*i]
        off += ls[2*i + 2] - ls[2*i + 1]
else:
    on = ls[1]
    for i in range(n // 2):
        on += ls[2*i + 3] - ls[2*i + 2]
        off += ls[2*i + 2] - ls[2*i + 1]
raw_on = on
for i in range(n // 2 + n % 2):
    on -= ls[2*i + 1] - ls[2*i]
    if ls[2*i + 2] - ls[2*i + 1] > 1:
        add.append(off - on - 1)
    off -= ls[2*i + 2] - ls[2*i + 1]
print(raw_on + max(add))
```

代码运行截图

The screenshot displays the Codeforces submission interface for problem 1000B, "Light It Up". The user's code is shown in a text editor, and the execution results are displayed below it. The code is a greedy solution that calculates the maximum time to keep the lights on by inserting a new light at the optimal position. The execution results show that the code passed all tests, with a verdict of "Accepted".

Test	Time	Memory	Exit Code	Checker Exit Code	Verdict
Test #1	15 ms	0 KB	0	0	OK
Test #2	30 ms	0 KB	0	0	OK

The submission was made by user xjsonghy on November 9, 2023, at 23:06 UTC+8. The code is written in Python 3 and is 666C - Number of Ways. The submission was accepted on test 2.

2. 学习总结和收获

部分题目以前做过，剩下的题目让我了解了一些新的类型的问题，学会了一些新的贪心/动态规划思路。

截至2023年11月21日，OJ完成题目110道，CF完成题目47道。