

# Assignment #B: 贪心、矩阵和动态规划

---

Updated 1710 GMT+8 Nov 21, 2023

2023 fall, Compiled by 钟明衡 物理学院

## 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\_NT x64 10.0.19045

Python编程环境: Visual Studio Code 1.76.1

C/C++编程环境: Visual Studio Code 1.76.1

## 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:

一开始直接用递推式写了一个，结果数字太大RE了

然后就直接用递推，没有超时

## 代码

```
1 t = int(input())
2 n = []
3 ans = [0, 1]
4 for _ in range(t):
5     n += [int(input())]
6 for i in range(1, max(n)):
7     ans += [(ans[-1]*2+ans[-2]) % 32767]
8 for nn in n:
9     print(ans[nn])
10
```

## 代码运行截图

### #42642126提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

#### 源代码

```
t = int(input())
n = []
ans = [0, 1]
for _ in range(t):
    n += [int(input())]
for i in range(1, max(n)):
    ans += [(ans[-1]*2+ans[-2]) % 32767]
for nn in n:
    print(ans[nn])
```

#### 基本信息

#: 42642126

题目: 02786

提交人: 23n2300011505(12号娱乐选

手)

内存: 40528kB

时间: 296ms

语言: Python3

提交时间: 2023-11-21 11:54:41

## 04133:垃圾炸弹

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

思路:

创建一个 $1025 \times 1025$ 的全为0的矩阵, 然后对于每一处垃圾, 把能炸到垃圾的位置加上垃圾数量

这样可以直接从最终的矩阵上读出最大值和最大值的个数, 即为答案

注意控制循环上下界不要越界了

## 代码

```
1 d = int(input())
2 n = int(input())
3 M = [[0]*1025 for i in range(1025)]
4 MAX = 0
5 maxx = 0
6 maxy = 0
7 ans = 0
```

```

8  for _ in range(n):
9      x, y, a = map(int, input().split())
10     maxx = max(x, maxx)
11     maxy = max(y, maxy)
12     for i in range(max(0, y-d), min(1024, y+d)+1):
13         for j in range(max(0, x-d), min(1024, x+d)+1):
14             M[i][j] += a
15             MAX = max(MAX, M[i][j])
16 for i in range(0, min(1024, maxy+d)+1):
17     for j in range(0, min(1024, maxx+d)+1):
18         if M[i][j] == MAX:
19             ans += 1
20 print('%d %d' % (ans, MAX))
21

```

代码运行截图

#42644168提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

d = int(input())
n = int(input())
M = [[0]*1025 for i in range(1025)]
MAX = 0
maxx = 0
maxy = 0
ans = 0
for _ in range(n):
    x, y, a = map(int, input().split())
    maxx = max(x, maxx)
    maxy = max(y, maxy)
    for i in range(max(0, y-d), min(1024, y+d)+1):
        for j in range(max(0, x-d), min(1024, x+d)+1):
            M[i][j] += a
            MAX = max(MAX, M[i][j])
for i in range(0, min(1024, maxy+d)+1):
    for j in range(0, min(1024, maxx+d)+1):
        if M[i][j] == MAX:
            ans += 1
print('%d %d' % (ans, MAX))

```

基本信息

#: 42644168

题目: 04133

提交人: 23n2300011505(12号娱乐选手)

内存: 11896kB

时间: 145ms

语言: Python3

提交时间: 2023-11-21 13:44:06

## 26971:分发糖果

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

思路:

之前写的思路:

由于只要求相邻rating大的小朋友糖果更多, 那么连续上升或下降的序列中, 最贪的方案是糖果数只±1

为了进一步地贪, 让每个单调序列“掉落”, 使得最小值都是1, 总和就是答案

这个思路代码太长了, 于是换了一个“掉落”的方法:

分别只顺序、倒序地判断是否上升，存储上升位置

比如输入为1 3 5 7 3 1，up为1 2 3 4 1 1，down为1 1 1 1 2 1

将up和down每个位置更大的那个加起来就是答案

### 代码

```
1 n = int(input())
2 l = list(map(int, input().split()))
3 ans = 0
4 up = [1]*n
5 down = [1]*n
6 for i in range(n-1):
7     if l[i+1] > l[i]:
8         up[i+1] = up[i]+1
9     if l[n-i-2] > l[n-i-1]:
10        down[n-i-2] = down[n-i-1]+1
11 for i in range(n):
12     ans += max(up[i], down[i])
13 print(ans)
14
```

### 代码运行截图

#### #42643204提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

#### 源代码

```
n = int(input())
l = list(map(int, input().split()))
ans = 0
up = [1]*n
down = [1]*n
for i in range(n-1):
    if l[i+1] > l[i]:
        up[i+1] = up[i]+1
    if l[n-i-2] > l[n-i-1]:
        down[n-i-2] = down[n-i-1]+1
for i in range(n):
    ans += max(up[i], down[i])
print(ans)
```

#### 基本信息

#: 42643204  
题目: 26971  
提交人: 23n2300011505(12号娱乐选手)  
内存: 5044kB  
时间: 39ms  
语言: Python3  
提交时间: 2023-11-21 13:12:00

## 26976:摆动序列

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

思路:

创建up、down两个列表，初始全为1

然后对序列的每一个元素 $a_i$ :

如果 $a_j < a_i, j < i$ , 则 $up[i] = \max(up[i], down[j] + 1)$

如果 $a_j > a_i, j < i$ , 则 $down[i] = \max(down[i], up[j] + 1)$

这样, up和down中最大的那个数就是答案

## 代码

```
1 n = int(input())
2 l = list(map(int, input().split()))
3 up = [1]*n
4 down = [1]*n
5 for i in range(1, n):
6     for j in range(0, i):
7         if l[i] > l[j]:
8             up[i] = max(up[i], down[j]+1)
9         elif l[i] < l[j]:
10            down[i] = max(down[i], up[j]+1)
11 print(max(max(up), max(down)))
12
```

## 代码运行截图

#42644453提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n = int(input())
l = list(map(int, input().split()))
up = [1]*n
down = [1]*n
for i in range(1, n):
    for j in range(0, i):
        if l[i] > l[j]:
            up[i] = max(up[i], down[j]+1)
        elif l[i] < l[j]:
            down[i] = max(down[i], up[j]+1)
print(max(max(up), max(down)))
```

基本信息

#: 42644453

题目: 26976

提交人: 23n2300011505(12号娱乐选手)

内存: 3656kB

时间: 221ms

语言: Python3

提交时间: 2023-11-21 13:59:19

## 27104:世界杯只因

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

思路:

在 $i$ 位置, 对于摄像头选择最优的情况是,  $i$ 在该摄像头 $j$ 的范围内, 且 $j$ 在右边能覆盖到最远的位置

因此, 递推式为 $i = \max(j + l[j]) + 1$ , 其中 $j$ 满足 $|j - i| \leq l[j]$

每次进行递推时,  $ans + 1$ , 退出条件为 $i \geq n$

最后输出 $ans$ 即可

## 代码

```
1 n = int(input())
2 l = list(map(int, input().split()))
3 i = 0
4 ans = 0
5 while i < n:
6     ans += 1
7     next = -1
8     for j in range(n):
9         if abs(j-i) <= l[j]:
10             next = max(next, j+l[j])
11     i = next+1
12 print(ans)
13
```

## 代码运行截图

#42645837提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
n = int(input())
l = list(map(int, input().split()))
i = 0
ans = 0
while i < n:
    ans += 1
    next = -1
    for j in range(n):
        if abs(j-i) <= l[j]:
            next = max(next, j+l[j])
    i = next+1
print(ans)
```

基本信息

#: 42645837

题目: 27104

提交人: 23n2300011505(12号娱乐选手)

内存: 62556kB

时间: 10575ms

语言: Python3

提交时间: 2023-11-21 15:32:11

## CF1000B: Light It Up

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

思路:

在间隔大于一的位置允许插入 $a$ , 插入会导致后面的亮暗互换

为了得到最长的亮的时间, 如果在亮的时间段插入, 就插在时间段的最后一个位置, 反之则插在时间段最前一个位置

在空格位置 $i$ 插入后的结果为(前面亮的总时间 + 后面暗的总时间 - 1)

用 $on$ 和 $off$ 来保存从头求和到现在的总亮、暗时间, 通过相减来得到一段的和

$maxans$ 初始为 $on[-1]$ , 如果 $i = 2k$ 或 $i = 2k + 1$ , 则后面的判断式为

$maxans = \max(maxans, on[2k] - 1 + off[-1] - off[2k])$

最终的 $maxans$ 就是答案

## 代码

```
1 n, m = map(int, input().split())
2 l = [0] + list(map(int, input().split()))+[m]
3 on = []
4 off = []
5 for i in range(n+1):
6     l[i] = l[i+1]-l[i]
7     if i % 2 == 0:
8         on.append(l[i])
9         off.append(0)
10    else:
11        on.append(0)
12        off.append(l[i])
13 for i in range(1, n+1):
14     on[i] += on[i-1]
15     off[i] += off[i-1]
16 maxans = on[-1]
17 for i in range(n+1):
18     if l[i] > 1:
19         maxans = max(maxans, on[i//2*2]-1+off[-1]-off[i//2*2])
20 print(maxans)
21
```

## 代码运行截图

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
233660833	Practice: MinghengZhong	<a href="#">1000B</a> - 27	Python 3	Accepted	186 ms	13460 KB	2023-11-21 11:42:50	2023-11-21 11:42:50	★	<button>Compare</button>

→ Source

Copy

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

## 2. 学习总结和收获

感觉这次的题目风格主要是：用递归或者简单的循环把一个东西存起来，要用的时候直接调用来节省时间

我觉得这次比上次的背包问题要简单一些，递归更好想，但是贪心依旧比较困难

