# Assignment #6: 矩阵、贪心

Updated 1432 GMT+8 Oct 14, 2025

2025 fall, Complied by 同学的姓名、院系

> **说明：**
>
> 1. **解题与记录：**
>
> 对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. 提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

# 1. 题目

## M18211: 军备竞赛

greedy, two pointers, http://cs101.openjudge.cn/pctbook/M18211

思路：

代码

```
#
p = int(input())
n = list(map(int, input().split()))
n.sort()
if p < n[0]:
    print(0)
else:
    s = sum(n)
    if p >= s:
        print(len(n))
```

```
    else:
        a = 0
        b = 0
        for i in range(len(n)-1, -1, -1):
            s -= n[i]
            if s <= p:
                a = i
                b = len(n)-a-1
                break
            p += n[i]
            if p >= s:
                a = i
                b = len(n)-a
                break
        print(a-b)
```

代码运行截图 （至少包含有"Accepted"）

#50363550提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
p = int(input())
n = list(map(int, input().split()))
n.sort()
if p < n[0]:
    print(0)
else:
    s = sum(n)
    if p >= s:
        print(len(n))
    else:
        a = 0
        b = 0
        for i in range(len(n)-1, -1, -1):
            s -= n[i]
            if s <= p:
                a = i
                b = len(n)-a-1
                break
            p += n[i]
            if p >= s:
                a = i
                b = len(n)-a
                break
        print(a-b)
```

基本信息

#: 50363550
题目: M18211
提交人: miko
内存: 3624kB
时间: 19ms
语言: Python3
提交时间: 2025-10-14 20:25:13

# M21554: 排队做实验

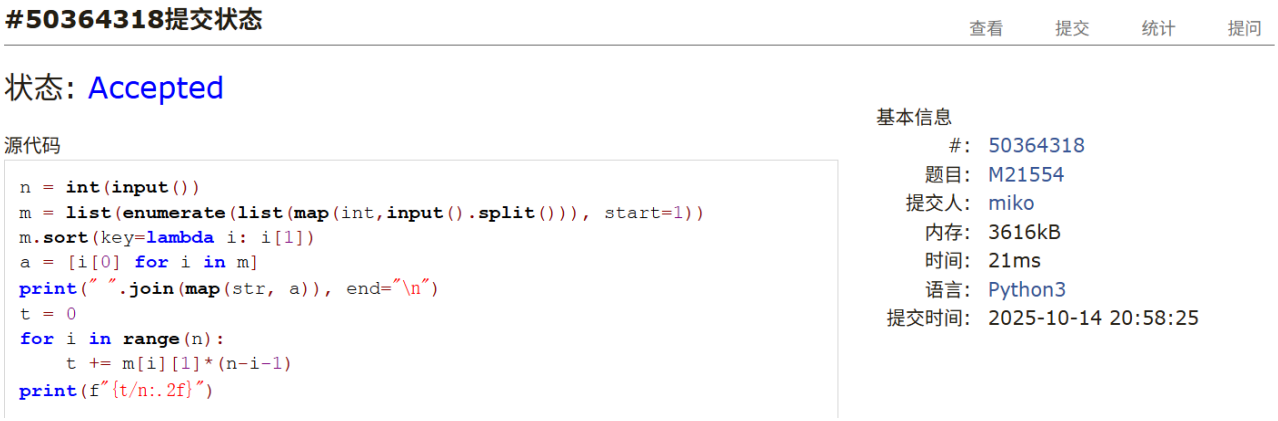
greedy, http://cs101.openjudge.cn/pctbook/M21554/

思路：

代码

```
n = int(input())
m = list(enumerate(list(map(int,input().split())), start=1))
m.sort(key=lambda i: i[1])
a = [i[0] for i in m]
print(" ".join(map(str, a)), end="\n")
t = 0
for i in range(n):
    t += m[i][1]*(n-i-1)
print(f"{t/n:.2f}")
```

代码运行截图 （至少包含有"Accepted"）




# E23555: 节省存储的矩阵乘法

implementation, matrices, http://cs101.openjudge.cn/pctbook/E23555

思路：

代码

```
n = list(map(int, input().split()))
m1 = []
m2 = []
m3 = []
for i in range(n[0]):
    m1.append([0]*n[0])
    m2.append([0]*n[0])
    m3.append([0]*n[0])
for i in range(n[1]):
    a, b , c = map(int, input().split())
    m1[a][b] = c
for i in range(n[2]):
    a, b, c = map(int, input().split())
    m2[a][b] = c
for i in range(n[0]):
```

```
    for j in range(n[0]):
        for k in range(n[0]):
            m3[i][j] += m1[i][k]*m2[k][j]
for i in range(n[0]):
    for j in range(n[0]):
        if m3[i][j] != 0:
            print(str(i)+" "+str(j)+" "+str(m3[i][j]))
```

代码运行截图 （至少包含有"Accepted"）

#50364766提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = list(map(int, input().split()))
m1 = []
m2 = []
m3 = []
for i in range(n[0]):
    m1.append([0]*n[0])
    m2.append([0]*n[0])
    m3.append([0]*n[0])
for i in range(n[1]):
    a, b , c = map(int, input().split())
    m1[a][b] = c
for i in range(n[2]):
    a, b, c = map(int, input().split())
    m2[a][b] = c
for i in range(n[0]):
    for j in range(n[0]):
        for k in range(n[0]):
            m3[i][j] += m1[i][k]*m2[k][j]
for i in range(n[0]):
    for j in range(n[0]):
        if m3[i][j] != 0:
            print(str(i)+" "+str(j)+" "+str(m3[i][j]))
```

基本信息

#: 50364766
题目: E23555
提交人: miko
内存: 3680kB
时间: 28ms
语言: Python3
提交时间: 2025-10-14 21:26:01

# M12558: 岛屿周长

matrices, http://cs101.openjudge.cn/pctbook/M12558

思路:

代码

```
#
n, m = map(int, input().split())
s = [[0]*(m+2)]
for i in range(1,n+1):
    s.append([0])
    s[i].extend(list(map(int,input().split())))
    s[i].append(0)
s.append([0]*(m+2))
c = 0
for i in range(1,n+1):
```

```
    for j in range(1,m+1):
        if s[i][j] == 0:
            continue
        else:
            c += 4
            if s[i-1][j] == 1:
                c -= 1
            if s[i][j-1] == 1:
                c -= 1
            if s[i+1][j] == 1:
                c -= 1
            if s[i][j+1] == 1:
                c -= 1
print(c)
```

代码运行截图 （至少包含有"Accepted"）

#50365052提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n, m = map(int, input().split())
s = [[0]*(m+2)]
for i in range(1,n+1):
    s.append([0])
    s[i].extend(list(map(int,input().split())))
    s[i].append(0)
s.append([0]*(m+2))
c = 0
for i in range(1,n+1):
    for j in range(1,m+1):
        if s[i][j] == 0:
            continue
        else:
            c += 4
            if s[i-1][j] == 1:
                c -= 1
            if s[i][j-1] == 1:
                c -= 1
            if s[i+1][j] == 1:
                c -= 1
            if s[i][j+1] == 1:
                c -= 1
print(c)
```

基本信息

#: 50365052
题目: M12558
提交人: miko
内存: 3604kB
时间: 21ms
语言: Python3
提交时间: 2025-10-14 21:46:02

# M01328: Radar Installation

greedy, http://cs101.openjudge.cn/practice/01328/

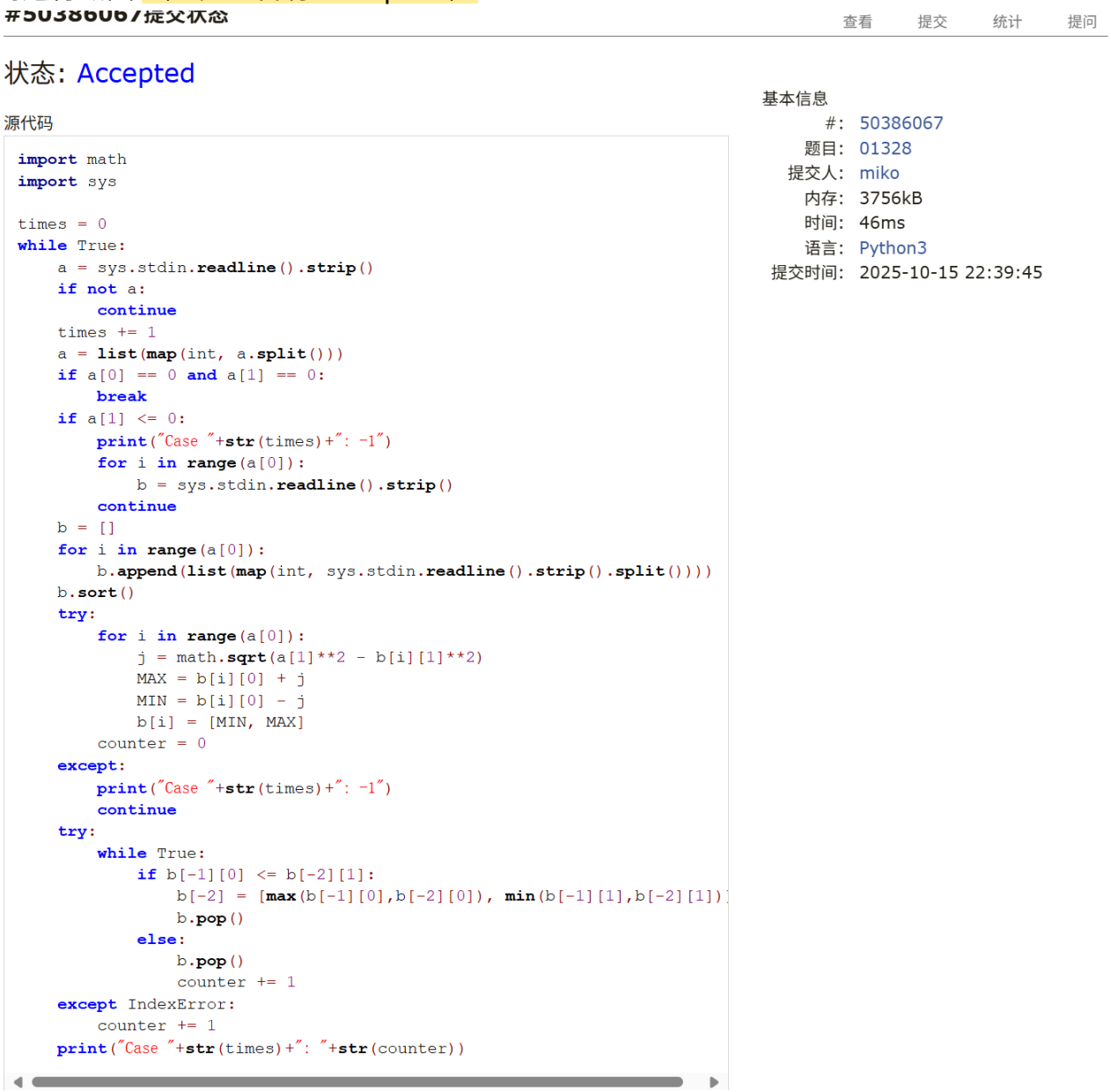
思路:

代码

```
import math
import sys
```

```
times = 0
while True:
    a = sys.stdin.readline().strip()
    if not a:
        continue
    times += 1
    a = list(map(int, a.split()))
    if a[0] == 0 and a[1] == 0:
        break
    if a[1] <= 0:
        print("Case "+str(times)+": -1")
        for i in range(a[0]):
            b = sys.stdin.readline().strip()
        continue
    b = []
    for i in range(a[0]):
        b.append(list(map(int, sys.stdin.readline().strip().split())))
    b.sort()
    try:
        for i in range(a[0]):
            j = math.sqrt(a[1]**2 - b[i][1]**2)
            MAX = b[i][0] + j
            MIN = b[i][0] - j
            b[i] = [MIN, MAX]
        counter = 0
    except:
        print("Case "+str(times)+": -1")
        continue
    try:
        while True:
            if b[-1][0] <= b[-2][1]:
                b[-2] = [max(b[-1][0],b[-2][0]), min(b[-1][1],b[-2][1])]
                b.pop()
            else:
                b.pop()
                counter += 1
    except IndexError:
        counter += 1
    print("Case "+str(times)+": "+str(counter))
```

代码运行截图 （至少包含有"Accepted"）



#50386067提交状态 查看 提交 统计 提问

状态：Accepted

源代码

```
import math
import sys

times = 0
while True:
    a = sys.stdin.readline().strip()
    if not a:
        continue
    times += 1
    a = list(map(int, a.split()))
    if a[0] == 0 and a[1] == 0:
        break
    if a[1] <= 0:
        print("Case "+str(times)+": -1")
        for i in range(a[0]):
            b = sys.stdin.readline().strip()
        continue
    b = []
    for i in range(a[0]):
        b.append(list(map(int, sys.stdin.readline().strip().split())))
    b.sort()
    try:
        for i in range(a[0]):
            j = math.sqrt(a[1]**2 - b[i][1]**2)
            MAX = b[i][0] + j
            MIN = b[i][0] - j
            b[i] = [MIN, MAX]
        counter = 0
    except:
        print("Case "+str(times)+": -1")
        continue
    try:
        while True:
            if b[-1][0] <= b[-2][1]:
                b[-2] = [max(b[-1][0],b[-2][0]), min(b[-1][1],b[-2][1])]
                b.pop()
            else:
                b.pop()
                counter += 1
    except IndexError:
        counter += 1
    print("Case "+str(times)+": "+str(counter))
```

基本信息

| | |
|---|---|
| #: | 50386067 |
| 题目: | 01328 |
| 提交人: | miko |
| 内存: | 3756kB |
| 时间: | 46ms |
| 语言: | Python3 |
| 提交时间: | 2025-10-15 22:39:45 |

# 545C. Woodcutters

dp, greedy, 1500, https://codeforces.com/problemset/problem/545/C

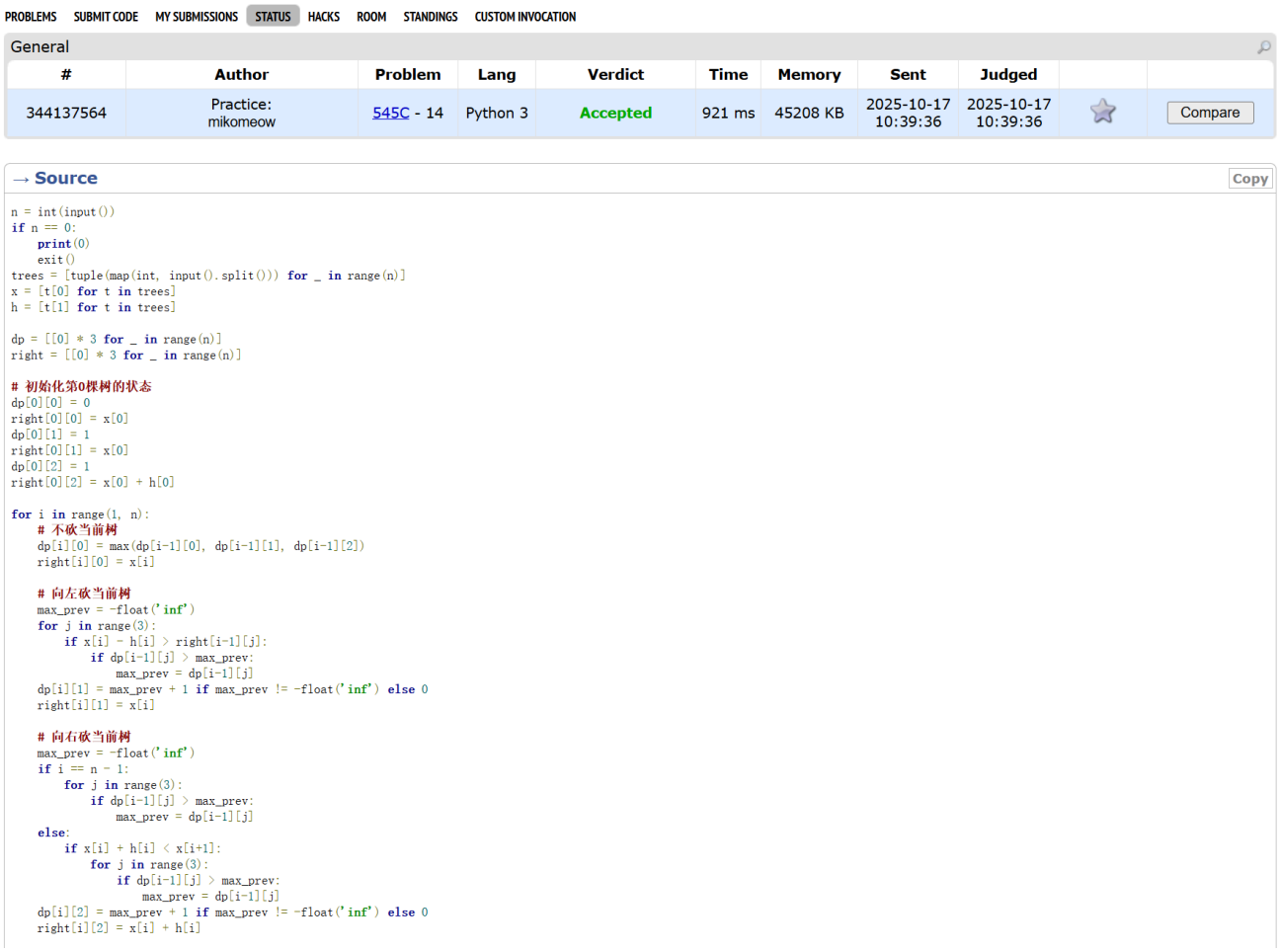
思路：

代码

```
n = int(input())
if n == 0:
    print(0)
    exit()
trees = [tuple(map(int, input().split())) for _ in range(n)]
x = [t[0] for t in trees]
h = [t[1] for t in trees]
```

```
dp = [[0] * 3 for _ in range(n)]
right = [[0] * 3 for _ in range(n)]

# 初始化第0棵树的状态
dp[0][0] = 0
right[0][0] = x[0]
dp[0][1] = 1
right[0][1] = x[0]
dp[0][2] = 1
right[0][2] = x[0] + h[0]

for i in range(1, n):
    # 不砍当前树
    dp[i][0] = max(dp[i-1][0], dp[i-1][1], dp[i-1][2])
    right[i][0] = x[i]

    # 向左砍当前树
    max_prev = -float('inf')
    for j in range(3):
        if x[i] - h[i] > right[i-1][j]:
            if dp[i-1][j] > max_prev:
                max_prev = dp[i-1][j]
    dp[i][1] = max_prev + 1 if max_prev != -float('inf') else 0
    right[i][1] = x[i]

    # 向右砍当前树
    max_prev = -float('inf')
    if i == n - 1:
        for j in range(3):
            if dp[i-1][j] > max_prev:
                max_prev = dp[i-1][j]
    else:
        if x[i] + h[i] < x[i+1]:
            for j in range(3):
                if dp[i-1][j] > max_prev:
                    max_prev = dp[i-1][j]
    dp[i][2] = max_prev + 1 if max_prev != -float('inf') else 0
    right[i][2] = x[i] + h[i]

print(max(dp[n-1][0], dp[n-1][1], dp[n-1][2]))
```

代码运行截图 （至少包含有"Accepted"）



PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

| # | Author | Problem | Lang | Verdict | Time | Memory | Sent | Judged | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 344137564 | Practice: mikomeow | 545C - 14 | Python 3 | Accepted | 921 ms | 45208 KB | 2025-10-17 10:39:36 | 2025-10-17 10:39:36 | ☆ | Compare |

→ Source Copy

```
n = int(input())
if n == 0:
    print(0)
    exit()
trees = [tuple(map(int, input().split())) for _ in range(n)]
x = [t[0] for t in trees]
h = [t[1] for t in trees]

dp = [[0] * 3 for _ in range(n)]
right = [[0] * 3 for _ in range(n)]

# 初始化第0棵树的状态
dp[0][0] = 0
right[0][0] = x[0]
dp[0][1] = 1
right[0][1] = x[0]
dp[0][2] = 1
right[0][2] = x[0] + h[0]

for i in range(1, n):
    # 不砍当前树
    dp[i][0] = max(dp[i-1][0], dp[i-1][1], dp[i-1][2])
    right[i][0] = x[i]

    # 向左砍当前树
    max_prev = -float('inf')
    for j in range(3):
        if x[i] - h[i] > right[i-1][j]:
            if dp[i-1][j] > max_prev:
                max_prev = dp[i-1][j]
    dp[i][1] = max_prev + 1 if max_prev != -float('inf') else 0
    right[i][1] = x[i]

    # 向右砍当前树
    max_prev = -float('inf')
    if i == n - 1:
        for j in range(3):
            if dp[i-1][j] > max_prev:
                max_prev = dp[i-1][j]
    else:
        if x[i] + h[i] < x[i+1]:
            for j in range(3):
                if dp[i-1][j] > max_prev:
                    max_prev = dp[i-1][j]
    dp[i][2] = max_prev + 1 if max_prev != -float('inf') else 0
    right[i][2] = x[i] + h[i]
```

# 2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2025fall每日选做”、CF、LeetCode、洛谷等网站题目。

额外题目：

## T27104:世界杯只因

查看 提交 统计 提问

总时间限制: 3600ms 单个测试点时间限制: 1200ms 内存限制: 131072kB

全局题号 **27104**
添加于 **2025-03-13**
提交次数 **208**
尝试人数 **47**
通过人数 **44**

你的提交记录

| # | 结果 | 时间 |
|---|---|---|
| 2 | Accepted | 2025-10-20 |
| 1 | Wrong Answer | 2025-10-20 |

### 描述

卡塔尔世界杯正在火热进行中，P大富哥李哥听闻有一种叫"肤白·态美·宇宙无敌·世界杯·预测鸡"的鸡品种（以下简称为只因）有概率能准确预测世界杯赛果，一口气买来无数只只因，并把它们塞进了N个只因窝里，但只因窝实在太多了，李哥需要安装摄像头来观测里面的只因的预测行为。

具体来说，李哥的只因窝可以看作分布在一条直线上的N个点，编号为1到N。由于每个只因窝的结构不同，在编号为i的只因窝处安装摄像头，观测范围为a_i，其中a是长为N的整数列，表示若在此安装摄像头，可以观测到编号在 [ i - a_i , i + a_i ]（闭区间）内的所有只因窝。

李哥觉得摄像头成本高，决定抠门一下，请你来帮忙看看最少需要安装多少个摄像头，才能观测到全部N个只因窝。作为回报，他会请你喝一杯芋泥波波牛乳茶。

### 输入

第一行：一个正整数，代表有N个只因窝。
第二行给出数列a：N个非负整数，第i个数代表a_i，也就是在第i个只因窝装摄像头能观测到的区间的半径。

数据保证 N ≤ 500000，0 ≤ a_i ≤ N

### 输出

一个整数，即最少需要装的摄像头数量。

### 样例输入

```
10
2 0 1 1 0 3 1 0 2 0
```

### 样例输出

```
3
```

### 提示

彩蛋：只因们很喜欢那个穿着蓝白球衣长得像黄金矿工的10号

### 来源

计概A 2022期末

## #50472198提交状态

查看 提交 统计 提问

状态：Accepted

基本信息
#: 50472198
题目: T27104
提交人: miko
内存: 108916kB
时间: 4406ms
语言: Python3
提交时间: 2025-10-20 20:26:30

源代码

```python
N = int(input())
n = list(map(int, input().split()))
m = []
r = 0
c = 1
for i in range(N):
        m.append([i-n[i],i+n[i]])
        if i-n[i] <= 0 and i+n[i] > r:
            r = i+n[i]
m.sort(key = lambda x: x[1],reverse = True)
while r < len(n)-1:
    for i in m:
        if i[0] <= r+1:
            c += 1
            r = i[1]
            break
print(c)
```


```python
N = int(input())
n = list(map(int, input().split()))
```

```
m = []
r = 0
c = 1
for i in range(N):
        m.append([i-n[i],i+n[i]])
        if i-n[i] <= 0 and i+n[i] > r:
            r = i+n[i]
m.sort(key = lambda x: x[1],reverse = True)
while r < len(n)-1:
    for i in m:
        if i[0] <= r+1:
            c += 1
            r = i[1]
            break
print(c)
```

## M12560:生存游戏

查看 提交 统计 提问

总时间限制: 1000ms 内存限制: 65536kB

全局题号 12560
添加于 2025-03-13
提交次数 49
尝试人数 43
通过人数 43

你的提交记录

| # | 结果 | 时间 |
|---|---|---|
| 1 | Accepted | 2025-10-17 |

**描述**

有如下生存游戏的规则:

给定一个n*m(1<=n,m<=100)的数组，每个元素代表一个细胞，其初始状态为活着(1)或死去(0)。<p="">

每个细胞会与其相邻的8个邻居（除数组边缘的细胞）进行交互，并遵守如下规则:

任何一个活着的细胞如果只有小于2个活着的邻居，那它就会由于人口稀少死去。

任何一个活着的细胞如果有2个或者3个活着的邻居，就可以继续活下去。

任何一个活着的细胞如果有超过3个活着的邻居，那它就会由于人口拥挤而死去。

任何一个死去的细胞如果有恰好3个活着的邻居，那它就会由于繁殖而重新变成活着的状态。

请写一个函数用来计算所给定初始状态的细胞经过一次更新后的状态是什么。

## #50420503提交状态

查看 提交 统计 提问

状态: Accepted

基本信息
#: 50420503
题目: M12560
提交人: miko
内存: 4100kB
时间: 36ms
语言: Python3
提交时间: 2025-10-17 20:54:03

源代码

```
n, m = map(int, input().split())
a = [[0]*(m+2)]
g = []
for i in range(1,n+1):
    b = list(map(int, input().split()))
    g.append(b)
    a.append([0])
    a[i].extend(b)
    a[i].append(0)
a.append([0]*(m+2))
for i in range(1,n+1):
    for j in range(1,m+1):
        k = a[i+1][j]+a[i-1][j]+a[i][j+1]+a[i][j-1]+a[i+1][j+1]+a[i+1][j
        if k == 2:
            pass
        elif k == 3:
            g[i-1][j-1] = 1
        else:
            g[i-1][j-1] = 0
for i in range(n):
    for j in range(m):
        print(g[i][j], end=' ')
    print()
```


```python
n, m = map(int, input().split())
a = [[0]*(m+2)]
g = []
for i in range(1,n+1):
    b = list(map(int, input().split()))
    g.append(b)
    a.append([0])
    a[i].extend(b)
    a[i].append(0)
a.append([0]*(m+2))
for i in range(1,n+1):
    for j in range(1,m+1):
```

```
        k = a[i+1][j]+a[i-1][j]+a[i][j+1]+a[i][j-1]+a[i+1][j+1]+a[i+1][j-
1]+a[i-1][j+1]+a[i-1][j-1]
        if k == 2:
            pass
        elif k == 3:
            g[i-1][j-1] = 1
        else:
            g[i-1][j-1] = 0
for i in range(n):
    for j in range(m):
        print(g[i][j], end=' ')
    print()
```

## 121. 买卖股票的最佳时机

已解答

简单 相关标签 相关企业

给定一个数组 `prices` ，它的第 `i` 个元素 `prices[i]` 表示一支给定股票第 `i` 天的价格。

你只能选择 **某一天** 买入这只股票，并选择在 **未来的某一个不同的日子** 卖出该股票。设计一个算法来计算你所能获取的最大利润。

返回你可以从这笔交易中获取的最大利润。如果你不能获取任何利润，返回 `0` 。


```
class Solution:

    def maxProfit(self, prices: List[int]) -> int:
```

```
        minnum = prices[0]
        maxnum = 0
        for i in prices:
            if i - minnum > maxnum:
                maxnum = i - minnum
            if i < minnum:
                minnum = i
        return maxnum
```

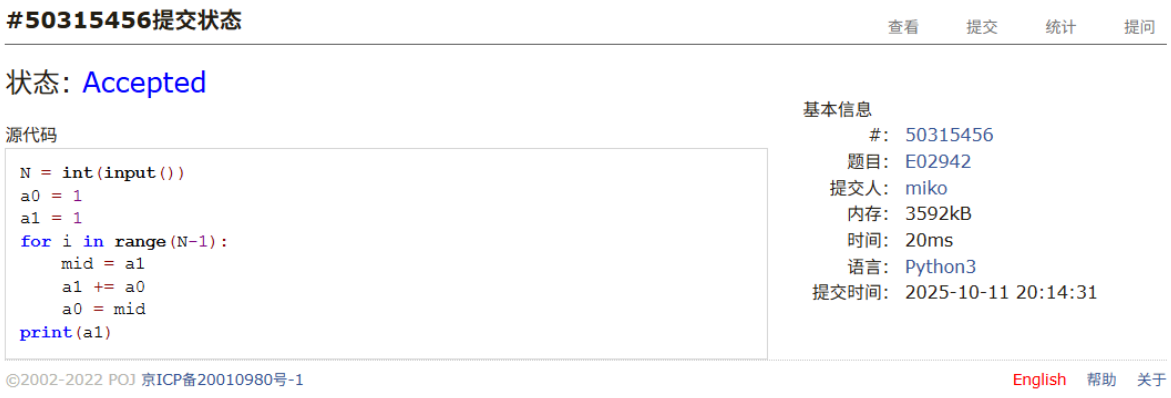
## E02942:吃糖果

总时间限制: 1000ms　内存限制: 65536kB

### 描述

名名的妈妈从外地出差回来，带了一盒好吃又精美的巧克力给名名（盒内共有 N 块巧克力，20 > N >0)。妈妈告诉名名每天可以吃一块或者两块巧克力。假设名名每天都吃巧克力，问名名共有多少种不同的吃完巧克力的方案。例如：如果N=1，则名名第1天就吃掉它，共有1种方案；如果N=2，则名名可以第1天吃1块，第2天吃1块，也可以第1天吃2块，共有2种方案；如果N=3，则名名第1天可以吃1块，剩2块，也可以第1天吃2块剩1块，所以名名共有2+1=3种方案；如果N=4，则名名可以第1天吃1块，剩3块，也可以第1天吃2块，剩2块，共有3+2=5种方案。现在给定N，请你写程序求出名名吃巧克力的方案数目。

### 输入




```
N = int(input())
a0 = 1
a1 = 1
for i in range(N-1):
    mid = a1
```

```
    a1 += a0
    a0 = mid
print(a1)
```

PROBLEMS SUBMIT STATUS STANDINGS CUSTOM TEST

## A. Cut Ribbon

time limit per test: 1 second
memory limit per test: 256 megabytes

Polycarpus has a ribbon, its length is $n$. He wants to cut the ribbon in a way that fulfils the following two conditions:

- After the cutting each ribbon piece should have length $a$, $b$ or $c$.
- After the cutting the number of ribbon pieces should be maximum.

Help Polycarpus and find the number of ribbon pieces after the required cutting.

### Input

The first line contains four space-separated integers $n$, $a$, $b$ and $c$ ($1 \le n, a, b, c \le 4000$) — the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly. The numbers $a$, $b$ and $c$ can coincide.

### Output

Print a single number — the maximum possible number of ribbon pieces. It is guaranteed that at least one correct ribbon cutting exists.

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

| # | Author | Problem | Lang | Verdict | Time | Memory | Sent | Judged | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 343086018 | Practice: mikomeow | 189A - 39 | Python 3 | Accepted | 77 ms | 60 KB | 2025-10-11 10:16:09 | 2025-10-11 10:16:09 | ☆ | Compare |

→ Source Copy

```
n, a, b, c = map(int, input().split())
dp = [-1] * (n + 1)
dp[0] = 0
for i in range(1, n + 1):
    if i >= a and dp[i - a] != -1:
        dp[i] = max(dp[i], dp[i - a] + 1)
    if i >= b and dp[i - b] != -1:
        dp[i] = max(dp[i], dp[i - b] + 1)
    if i >= c and dp[i - c] != -1:
        dp[i] = max(dp[i], dp[i - c] + 1)
print(dp[n])
```

```
n, a, b, c = map(int, input().split())
dp = [-1] * (n + 1)
dp[0] = 0
for i in range(1, n + 1):
    if i >= a and dp[i - a] != -1:
        dp[i] = max(dp[i], dp[i - a] + 1)
    if i >= b and dp[i - b] != -1:
        dp[i] = max(dp[i], dp[i - b] + 1)
    if i >= c and dp[i - c] != -1:
        dp[i] = max(dp[i], dp[i - c] + 1)
print(dp[n])
```

1.无
2.用enumerate跟踪元素位置
3.矩阵乘法
4.无

5.很难评价的一道题，题中确实没有说d>=0，但我无法想象出探测范围是负数的雷达:)，看了测试用例才做出来
6.使用了ai，这道题难点在于初始状态以及dp的思路