

A - Triangular Number 三角数

题面翻译

时间限制：2 秒 / 内存限制：1024 MiB
分数：100 分

题目描述

给定一个正整数 N 。
输出从 1 到 N 的所有整数的和，即 $1 + 2 + \cdots + N$ 。

约束条件

- $1 \leq N \leq 100$
- N 是整数

输入格式

输入从标准输入按以下格式给出：

N

输出格式

输出从 1 到 N 的所有整数的和。

样例输入 1

5

样例输出 1

15

样例解释 1

因为 $1 + 2 + 3 + 4 + 5 = 15$ ，所以输出 15。

样例输入 2

1

样例输出 2

1

样例输入 3

29

样例输出 3

435

思考

很简单，方法在题面中已经给出了，即 $\sum_{i=1}^n i$ ，这个算法的时间复杂度是 $O(n)$ 的。

也可以使用高斯求和的方法， $\frac{n(1+n)}{2}$ 。

我太懒了，使用高斯求和的方法。

代码

```
#include <bits/stdc++.h>
using namespace std;
int n,x,i,s;
int main(){
    cin>>n;
    for(i=1;i<=n;i++) s+=i;
    cout<<s;
}
```

B - No-Divisible Range 无因数区间

题目翻译

时间限制：2 秒 / 内存限制：1024 MiB

分数：200 分

题目描述

给定一个长度为 N 的正整数序列 $A = (A_1, A_2, \dots, A_N)$ 。

请找出满足 $1 \leq l \leq r \leq N$ 的整数对 (l, r) 的数量，要求这些整数对满足如下条件：

对于所有满足 $l \leq i \leq r$ 的整数 i ， A_i **不是** 区间 $[l, r]$ 内所有元素的和的约数。

约束条件

- $1 \leq N \leq 50$
- $1 \leq A_i \leq 1000$
- 所有输入值均为整数

输入格式

输入数据通过标准输入给出，格式如下：

```
N
A_1 A_2 ... A_N
```

输出格式

输出满足条件的整数对 (l, r) 的数量。

样例输入 1

```
5
8 6 10 5 7
```

样例输出 1

```
6
```

样例解释 1

序列 $A = (8, 6, 10, 5, 7)$ 。

例如，整数对 $(l, r) = (1, 2)$ 满足条件：区间和为 $A_1 + A_2 = 14$ ， $A_1 = 8$ 和 $A_2 = 6$ 都不是 14 的约数。

而整数对 $(l, r) = (1, 3)$ 不满足条件：区间和为 $A_1 + A_2 + A_3 = 24$ ， $A_1 = 8$ 是 24 的约数。

满足条件的整数对为 $(1, 2), (1, 4), (2, 3), (2, 4), (3, 5), (4, 5)$ ，共 6 个，因此输出 6。

样例输入 2

```
3
1 1 1
```

样例输出 2

```
0
```

思考

就不打炮打蚊子了。

直接暴力枚举 l 到 r ，求和为 S ，公式为 $S = \sum_{i=l}^r a_i$ ，然后判断从 l 到 r 如果能被 S 整除，那这个就不是一个无因数区间，不应该计入到答案，否则就计入到答案，最后输出即可。

警告：暴力枚举 l 到 r 的时候，需要排除 $l > r$ 的状态，即 `if(l>r) continue;`。

时间复杂度为 $O(n^3)$ ，题目规定 $N \leq 50$ ，可以通过。

代码

```
#include <bits/stdc++.h>
using namespace std;
int n,i,a[60],f1,s,t,w,ans;
int main(){
    cin>>n;
    for(i=1;i<=n;i++) cin>>a[i];
    for(t=1;t<=n;t++) for(w=1;w<=n;w++){
        if(t>w) continue;
        f1=1;s=0;
        for(i=t;i<=w;i++) s+=a[i];
        for(i=t;i<=w;i++) if(s%a[i]==0) f1=0;
        ans+=f1;
    }
    cout<<ans;
}
```

C - Domino 多米诺

题面翻译

时间限制：2 秒 / 内存限制：1024 MiB

分数：300 分

题目描述

有 N 张多米诺骨牌排成一行，放置在数轴上。第 i 张骨牌位于坐标 i 处，高度为 A_i 。

当第 i 张骨牌向右倒下时，坐标在 i 到 $i + A_i - 1$ （包含两端）范围内的所有骨牌都会向右倒下。

当第一张骨牌向右倒下时，总共会有多少张骨牌倒下？

约束条件

- $1 \leq N \leq 5 \times 10^5$
- $1 \leq A_i \leq N$
- 所有输入值均为整数

输入格式

输入数据通过标准输入给出，格式如下：

```
N
A_1 A_2 ... A_N
```

输出格式

输出当第一张骨牌向右倒下时，倒下的骨牌总数。

样例输入 1

```
4
3 1 4 1
```

样例输出 1

```
4
```

样例解释 1

当第一张骨牌向右倒下时，第二张和第三张骨牌也会向右倒下。当第三张骨牌向右倒下时，第四张骨牌也会倒下。

样例输入 2

```
9
1 4 1 4 2 1 3 5 6
```

样例输出 2

```
1
```

样例解释 2

当第一张骨牌向右倒下时，没有其他骨牌会倒下。

样例输入 3

```
10
5 4 3 2 1 1 2 3 4 5
```

样例输出 3

```
5
```

思考

数据范围 $N \leq 5 \times 10^5$ ，暴力模拟每张骨牌的连锁倒下会超时，不能用 $O(N^2)$ 的暴力枚举了，必须用 $O(N)$ 贪心策略。

核心是维护**当前最远覆盖位置**和**遍历边界**：

1. 第一张骨牌倒下的初始覆盖范围是 $1 + A[1] - 1$ ，记为 `ma`，用 `tmp` 记录当前需要遍历的右边界；
2. 从第 2 张骨牌开始，在 `tmp` 范围内遍历，不断更新 `ma` 为所有骨牌能覆盖的最远位置；
3. 遍历到 `tmp` 时，将 `tmp` 更新为新的 `ma`，扩展遍历范围；
4. 最终答案取 `ma` 和 N 的较小值，避免覆盖范围超出骨牌总数。

小声：我看到时间限制为 2s 的时候以为不用贪心，就用暴力枚举，结果 TLE 了我操，∵ 数据范围是指数级的，是 2.5×10^{11} ，但实际上计算机每秒大约只能做 10^9 次，∴ 会超时。

代码

```
#include <bits/stdc++.h>
using namespace std;
const int N=500010;
int a[N],n,i,ma,tmp;
int main(){
    cin>>n;
    for(i=1;i<=n;i++) cin>>a[i];
    if(n==0) return cout<<0,0;
    ma=1+a[1]-1;
    tmp=ma;
    i=2;
    while(i<=tmp&& i<=n){
        if(i+a[i]-1>ma) ma=i+a[i]-1;
        if(i==tmp) tmp=ma;
        i++;
    }
    if(ma>n) cout<<n;
    else cout<<ma;
}
```

D - Reachability Query 2 可达性查询 2

题面翻译

时间限制：3 秒 / 内存限制：1024 MiB

分数：425 分

题目描述

给定一个有 N 个顶点、 M 条边的有向图。顶点编号为 1 到 N ，第 i 条边是从顶点 X_i 指向顶点 Y_i 的有向边。初始时，所有顶点均为白色。

按顺序处理 Q 个查询，每个查询为以下两种类型之一：

- 1 v ：将顶点 v 染成黑色。
- 2 v ：判断从顶点 v 出发，沿着边遍历是否能够到达某个黑色顶点。

约束条件

- $1 \leq N \leq 3 \times 10^5$
- $0 \leq M \leq 3 \times 10^5$
- $1 \leq Q \leq 3 \times 10^5$
- $1 \leq X_i, Y_i \leq N$
- 图中无自环（即 $X_i \neq Y_i$ ）。
- 图中无重边（即所有 (X_i, Y_i) 互不相同）。
- 查询中， $1 \leq v \leq N$ 。
- 所有输入值均为整数。

输入格式

输入数据通过标准输入给出，格式如下：

```
N M
X_1 Y_1
⋮
X_M Y_M
Q
query_1
⋮
query_Q
```

其中 `query_i` 表示第 i 个查询，格式为以下两者之一：

```
1 v
2 v
```

输出格式

设第二类查询的数量为 q ，输出 q 行结果：

- 对于第 i 个第二类查询，若从顶点 v 能到达黑色顶点，输出 `Yes`；否则输出 `No`。

样例输入 1

```
5 6
1 2
2 3
3 1
4 5
1 4
2 5
5
1 3
2 1
2 4
1 5
2 4
```

样例输出 1

```
Yes
No
Yes
```

样例解释

初始时，给定的图如最左侧的图所示。

第一个查询将顶点 3 染成黑色（如中间的图所示）。

第二个查询中，从顶点 1 可以到达黑色顶点 3，故输出 `Yes`。

第三个查询中，从顶点 4 无法到达任何黑色顶点，故输出 `No`。

第四个查询将顶点 5 染成黑色（如最右侧的图所示）。

第五个查询中，从顶点 4 可以到达黑色顶点 5，故输出 `Yes`。

思考

很明显，这是一道图论题，其实我个人做的图论题也不多，我首先感觉是 [P3916 图的遍历 - 洛谷](#) 的升级版，多了染色和一些杂七杂八的东西，抽象。

小声：我刚看到每个查询的时候我还以为他考的是并查集。

一个经典的解决动态图可达性问题的方法是使用 `bitset` 或 分块，但 $N = 3 \times 10^5$ ，`bitset` 太大（ $N = 3 \times 10^5$ bits 约 37.5 KB 每个顶点，总共 $N = 3 \times 10^5$ 个顶点就是 10^{10} bits ≈ 1.25 GB，太特么大了）。

我草比赛怎么结束了？

E - Cover query 覆盖查询

题面翻译

时间限制：2 秒 / 内存限制：1024 MiB

分数：450 分

题目描述

有 N 个单元格从左到右排成一行。

从左数第 i 个单元格 ($1 \leq i \leq N$) 被称为单元格 i 。

初始时，所有单元格均为白色。

按顺序处理 Q 个查询。

第 i 个查询 ($1 \leq i \leq Q$) 的内容如下：

给定两个整数 L_i 和 R_i ($1 \leq L_i \leq R_i \leq N$)。

将单元格 $L_i, L_i + 1, \dots, R_i$ 全部染成黑色。

在此操作中，原本白色的单元格会被染成黑色，原本黑色的单元格保持黑色不变。

操作完成后，求出 N 个单元格中仍为白色的单元格数量。

约束条件

- $1 \leq N \leq 10^9$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq L_i \leq R_i \leq N$
- 所有输入值均为整数

输入格式

输入数据通过标准输入按以下格式给出：

```
N Q
L_1 R_1
L_2 R_2
⋮
L_Q R_Q
```

输出格式

输出 Q 行。
第 i 行 ($1 \leq i \leq Q$) 输出第 i 个查询的答案。

样例输入 1

```
10 5
3 5
8 9
5 8
2 9
6 6
```

样例输出 1

```
7
5
3
2
2
```

样例解释

初始时，10 个单元格从左到右排成一行。

第一个查询将单元格 3、4、5 染成黑色。操作后，白色单元格为 1、2、6、7、8、9、10，共 7 个。

第二个查询将单元格 8、9 染成黑色。操作后，白色单元格为 1、2、6、7、10，共 5 个。

第三个查询将单元格 5、6、7、8 染成黑色。操作后，白色单元格为 1、2、10，共 3 个。

第四个查询将单元格 2 到 9 染成黑色。操作后，白色单元格为 1、10，共 2 个。

第五个查询将单元格 6 染成黑色。操作后，白色单元格仍为 1、10，共 2 个。

因此，按顺序输出 7、5、3、2、2。

样例输入 2

```
1000000000 1
1 500000000
```

样例输出 2

500000000

思考

[待补]

F - Cat exercise 猫咪的锻炼

题面翻译

时间限制：2 秒 / 内存限制：1024 MiB

分数：550 分

题目描述

有 N 座猫塔从左到右排成一行，从左数第 i 座猫塔 ($1 \leq i \leq N$) 的高度为 P_i 。

其中，序列 $(P_1, P_2, \dots, P_N)$ 是 $(1, 2, \dots, N)$ 的一个排列。

从左数第 i 座猫塔和第 j 座猫塔之间的距离定义为 $|i - j|$ 。

初始时，有一只猫位于高度为 N 的猫塔顶端。

高桥想要锻炼这只猫，方式是**重复选择并移除猫塔**。

当高桥移除一座猫塔时，猫的移动规则如下：

1. 若猫**不在**被移除的猫塔顶端，则猫保持不动。
2. 若猫**在**被移除的猫塔顶端，且该猫塔的左侧或右侧至少有一座猫塔存在，则猫会移动到以下目标猫塔：
 - 目标猫塔的范围是：从被移除的猫塔出发，通过**重复移动到相邻猫塔**能够到达的所有猫塔（不包含被移除的猫塔）。
 - 目标猫塔是上述范围内**高度最高**的猫塔。
 - 此时，猫的移动距离计入总距离，该距离等于移动前所在猫塔和移动后所在猫塔之间的距离（与移动过程中经过的猫塔高度、数量无关）。
3. 若猫**在**被移除的猫塔顶端，且该猫塔的左右两侧都没有猫塔存在，则猫会跳进高桥的怀里，锻炼就此结束。这种情况下，不计入任何移动距离。

注意：被移除的猫塔留下的空位不会被填补。也就是说，初始时从左数第 i 座猫塔的相邻猫塔仅为第 $i - 1$ 和第 $i + 1$ 座猫塔（若存在），后续不会因为其他猫塔被移除而与非原本相邻的猫塔变成相邻关系。

请你求出，锻炼结束前猫能够移动的**最大可能总距离**。

约束条件

- $1 \leq N \leq 2 \times 10^5$
- $(P_1, P_2, \dots, P_N)$ 是 $(1, 2, \dots, N)$ 的一个排列
- 所有输入值均为整数

输入格式

输入数据通过标准输入按以下格式给出：

```
N
P_1 P_2 ... P_N
```

输出格式

输出锻炼结束前猫能够移动的最大可能总距离。

样例输入 1

```
5
5 3 4 1 2
```

样例输出 1

```
5
```

样例解释 1

初始时，猫塔的高度从左到右依次为 5、3、4、1、2。

下文将从左数第 i 座猫塔 ($1 \leq i \leq 5$) 简称为猫塔 i 。

初始时，猫位于猫塔 1 的顶端（高度为 5）。

若高桥按照 $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4$ 的顺序移除猫塔，猫的移动过程如下：

- 1. 移除猫塔 1：从猫塔 1 出发，通过重复移动到相邻猫塔可到达的猫塔为 2、3、4、5（不含猫塔 1）。其中高度最高的是猫塔 3（高度为 4），猫移动到该猫塔。
- 2. 移除猫塔 2：猫不在该猫塔顶端，保持不动。
- 3. 移除猫塔 3：从猫塔 3 出发，可到达的猫塔为 4、5（不含猫塔 3）。其中高度最高的是猫塔 5（高度为 2），猫移动到该猫塔。
- 4. 移除猫塔 5：从猫塔 5 出发，可到达的猫塔仅有 4（不含猫塔 5），猫移动到该猫塔。
- 5. 移除猫塔 4：猫的左右两侧都没有猫塔，跳进高桥怀里，锻炼结束。

此时，猫的总移动距离为 $|1 - 3| + |3 - 5| + |5 - 4| = 5$ 。不存在任何一种猫塔移除顺序，能让猫的总移动距离达到 6 或以上，因此输出 5。

样例输入 2

```
3
1 3 2
```

样例输出 2

1

样例解释 2

初始时，猫塔的高度从左到右依次为 1、3、2。

下文将从左数第 i 座猫塔 ($1 \leq i \leq 3$) 简称为猫塔 i 。

初始时，猫位于猫塔 2 的顶端（高度为 3）。

若高桥按照 $2 \rightarrow 3$ 的顺序移除猫塔，猫的移动过程如下：

1. 移除猫塔 2：从猫塔 2 出发，可到达的猫塔为 1、3（不含猫塔 2）。其中高度最高的是猫塔 3（高度为 2），猫移动到该猫塔。
2. 移除猫塔 3：猫的左右两侧都没有猫塔，跳进高桥怀里，锻炼结束。

此时，猫的总移动距离为 $|2 - 3| = 1$ ，这是最大可能的总距离。

注意：移除猫塔 2 后，猫塔 1 和猫塔 3 **不会被视为相邻**。

思考

[待补]

G - Domino Arrangement 多米诺序列

题面翻译

时间限制：2 秒 / 内存限制：1024 MiB

分数：600 分

题目描述

有 N 个单元格从左到右依次编号为 1 到 N 。初始时，所有单元格均未被涂上任何颜色。

共有 M 种颜色，对于第 i 种颜色，你可以选择单元格区间 $[L_i, R_i]$ （即 $L_i, L_i + 1, \dots, R_i$ ）内**任意数量**的单元格，将其涂为该颜色（可选择涂 0 个、1 个或多个）。

请计算满足以下条件的涂色方案数，并将结果对 998244353 取模：

- 对于每一个单元格 i ：
 - 若该单元格被涂上了某一种颜色，则其左侧相邻单元格 $i - 1$ 和右侧相邻单元格 $i + 1$ 中，**恰好有一个** 单元格被涂成了与 i 相同的颜色；
 - 特别说明：单元格 0 和 $N + 1$ 被视为未被任何颜色涂色（即不存在的单元格）。

约束条件

- $1 \leq N, M \leq 5 \times 10^5$
- $1 \leq L_i \leq R_i \leq N$
- 所有输入值均为整数

输入格式

输入数据从标准输入按以下格式给出：

```
N M
L_1 R_1
L_2 R_2
⋮
L_M R_M
```

输出格式

输出满足条件的涂色方案数（对 998244353 取模）。

样例输入 1

```
5 2
1 3
1 5
```

样例输出 1

```
11
```

样例解释 1

第 1 种颜色可涂色的单元格范围是 1, 2, 3, 第 2 种颜色可涂色的单元格范围是 1, 2, 3, 4, 5。
满足条件的涂色方案共有 11 种。

样例输入 2

```
3 3
1 1
2 2
3 3
```

样例输出 2

```
1
```

样例解释 2

唯一满足条件的方案是：不涂任何单元格（若给任意单元格涂色，都无法满足“恰好一个相邻单元格同色”的条件）。

样例输入 3

```
500000 10
1 499999
2 499998
3 499997
4 499996
5 499995
6 499994
7 499993
8 499992
9 499991
10 499990
```

样例输出 3

```
775503999
```

补充说明

答案需对 998244353 取模后输出。

思考

[待补]