NOIP2020 模拟赛 Day1

出题学校：长乐一中

题目名称	魔法	连通性	矩形	排列
题目类型	传统型	传统型	传统型	传统型
提交程序名	`magic.cpp`	`floyd.cpp`	`rectangle.cpp`	`permutation.cpp`
输入文件名	`magic.in`	`floyd.in`	`rectangle.in`	`permutation.in`
输出文件名	`magic.out`	`floyd.out`	`rectangle.out`	`permutation.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
空间限制	512 MB	512 MB	512 MB	512 MB
子任务数目	6	6	6	10
测试点是否等分	否	否	否	是
是否有 `Special Judge`	是	否	否	否

注意事项

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. $\mathrm{C}++$ 中函数 $\mathrm{main}()$ 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. **提交文件请对每道题目分别建立子文件夹，仅支持 $\mathrm{C}++$ 语言。**
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
6. 只提供 Linux 格式附加样例文件。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. **最终评测时的编译选项为** `-O2 -lm -std=c++11`。
9. 最终评测配置为：Intel(R) Core(TM) i5-9500 CPU @3.00GHz，内存 8GB，上述时限以此配置为准。
10. 评测系统为 Ubuntu 18.04 LTS x64，编译器版本为 G++ 7.5.0。

其他说明

1. 每题的时间限制均至少为标准程序在与评测相同的环境下最慢运行时间的 2 倍。
2. 题目难度和顺序**不一定**有直接关系。
3. 比赛时间为 4 小时。

A. 魔法（magic）

题目描述

有 n 个小球排成一行，每个小球是红色或者蓝色。开始你被给定了两个非负整数 R 和 B。

每次你可以施展一个魔法，每次魔法你可以选择**任意不同的 $R+B$ 个球**，将这些球全部变成白色，但是需要满足下列条件：

- 你选择的 $R+B$ 个球中，需要有恰好 R 个红球和恰好 B 个蓝球。
- 你选择的 $R+B$ 个球中，最左端的球和最右端的球之间，不能有之前已经变成白色的球。

问是否能用若干次魔法将所有球都变成白色，若可以请输出**任意一组方案**。

输入格式

从文件 `magic.in` 中读入数据。

第一行三个整数 n, R, B，含义见题目描述。

第二行一个长度为 n 的只含 `R` 或 `B` 的字符串，第 i 个字符表示第 i 个球的颜色，`R` 表示红色，`B` 表示蓝色。

输出格式

输出到文件 `magic.out` 中。

如果不存在合法方案，你只要输出一行 `NO` 即可。

否则如果存在合法方案，你需要在第一行输出一个 `YES`，在第二行输出一个整数 S 表示使用魔法的次数。
接下来 S 行，每行 $R+B$ 个整数，表示每一次魔法选中的球的编号。

请注意，每一行内的编号可以按照任意顺序，但是**这 S 行需要按照时间顺序从前到后输出**。如果有多组合法解，你只要输出任意一组即可。

样例

样例输入 1

```
6 1 1
RBBRRB
```

样例输出 1

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

样例输出 2

```
8 1 2
BRBRBRBR
```

样例输出 2

```
NO
```

数据规模与约定

本题采用子任务捆绑测试。对于每个子任务，你只有通过了这个子任务的所有数据，才能获得这个子任务的分数。

- 子任务 1（10 分）：$R=0$；
- 子任务 2（15 分）：$R=B=1$；
- 子任务 3（15 分）：$n\le 10$；
- 子任务 4（20 分）：$n\le 200$；
- 子任务 5（15 分）：$n\le 2000$；
- 子任务 6（25 分）：无特殊限制。

对于所有数据，$R,B\ge 0$，$1\le R+B\le n\le 10^5$。

B. 连通性（floyd）

题目描述

对于一个点数为 n 的简单无向图（无重边无自环），我们可以用一个二维布尔数组 `g[][]` 来描述这个无向图：当且仅当图中存在边 (u, v) 时 `g[u][v]=true`，否则 `g[u][v]=false`。

接下来我们可以用 Floyd 算法来求出这张图的连通性，代码如下：

```
void Floyd(int n)
{
    for (int k = 1; k <= n; ++k)
        for (int i = 1; i <= n; ++i)
            for (int j = 1; j <= n; ++j)
                g[i][j] = g[i][j] || (i != j && g[i][k] && g[k][j]);
}
```

执行完 `Floyd(n)` 之后，我们就得到了二维布尔数组 `g[][]` 来描述这个无向图的连通性：当且仅当图中存在 u, v 之间的路径时 `g[u][v]=true`，否则 `g[u][v]=false`。

有一天，执行这个算法的电脑被入侵了，这个算法被篡改成了下面这个代码：

```
void AttackedFloyd(int n, int m)
{
    for (int k = 1; k <= n - m; ++k)
        for (int i = 1; i <= n; ++i)
            for (int j = 1; j <= n; ++j)
                g[i][j] = g[i][j] || (i != j && g[i][k] && g[k][j]);
}
```

现在给定 n, m，你需要求出有多少不同的点数为 n 的简单无向图，使得执行 `AttackedFloyd(n, m)` 和执行 `Floyd(n)` 得到的二维数组 `g[][]` 相同。答案对 $10^9 + 7$ 取模。

两个简单无向图不同，当且仅当存在点对 u, v，使得其中一张无向图中存在边 (u, v) 而另一张无向图中不存在。

输入格式

从文件 `floyd.in` 中读入数据。

本题的每个测试点包含多组询问。

第一行包含一个整数 T 表示询问组数。

接下来 T 行，每行两个整数 n, m 表示一组询问。

输出格式

输出到文件 `floyd.out` 中。

对于每组询问，输出一行一个整数表示答案对 $10^9 + 7$ 取模的结果，共输出 T 行。

样例

样例输入

```
8
1 1
2 2
3 1
10 0
10 1
10 8
10 9
10 10
```

样例输出

```
1
2
7
371842544
598590656
929725695
302765502
115975
```

数据规模与约定

本题采用子任务捆绑测试。对于每个子任务，你只有通过了这个子任务的所有数据，才能获得这个子任务的分数。

- 子任务 1（20 分）：$n \le 5$；
- 子任务 2（20 分）：$n \le 15$；
- 子任务 3（15 分）：$m = 0$；
- 子任务 4（15 分）：$m = 1$；
- 子任务 5（15 分）：$m = n$；
- 子任务 6（15 分）：无特殊限制。

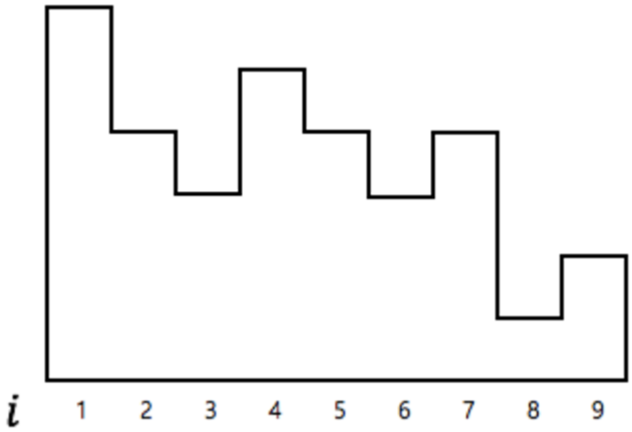
对于所有数据，$1 \le n \le 100$，$1 \le T \le 10^5$，$0 \le m \le n$。

C. 矩形（rectangle）

题目描述

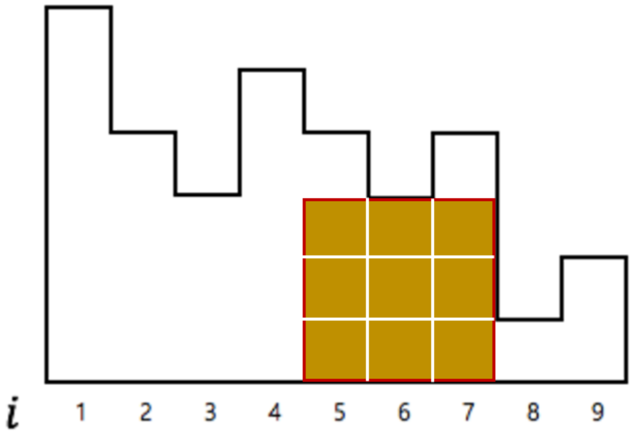
直方图是将 n 个底边在同一直线的相邻矩形对齐而成的多边形。每个长方形被称为条形。从左到右第 i 条的宽度为 1，高度为 h_i。

下图就描述了 $n=9$，$h=\{6,4,3,5,4,3,4,1,2\}$ 的情况。



有一天，Bob 想要找到在这个直方图中，面积最大的，和直方图的边平行的矩形。他想到了一种暴力做法：

- 初始有一个空数组 A。
- 对于每个满足 $1 \le i \le j \le n$ 的整数对 (i,j)，找到面积最大的矩形，使得这个矩形的底边为第 $i \sim j$ 个条形的底边且矩形在直方图内部，然后将这个矩形的面积加入到数组 A 中。
  形式化地，即把 $(j-i+1) \times \min_{k=i}^{j} h_k$ 加入数组 A 中。
  如下图，当 $i=5, j=7$ 时，这个矩形的面积为 9。



- 最后会得到一个长度为 $\frac{n(n+1)}{2}$ 的数组 A。将数组 A 按照不降序排序，那么数组 A 的最后一个元素即为答案。

由于 Bob 太懒了，于是他给了你 n 和数组 h，他想让你帮他求出**按照不降序排序**后的数组 A。

由于 A 数组可能很长，你只需要求出数组 A 的某一段即可。具体地，Bob 会给你两个正整数 $L,R(L \le R)$。你只需要求出 $A_L, A_{L+1}, \ldots, A_{R-1}, A_R$。

输入格式

从文件 `rectangle.in` 中读入数据。

第一行包含一个正整数 n，表示数组 h 的长度。

第二行包含 n 个正整数，第 i 个正整数表示 h_i。

第三行包含两个正整数 L, R。

输出格式

输出到文件 `rectangle.out` 中。

输出一行 $R - L + 1$ 个整数，依次表示 $A_L, A_{L+1}, \ldots, A_{R-1}, A_R$，相邻两个整数之间用一个空格隔开。

样例

样例输入 1

```
9
6 4 3 5 4 3 4 1 2
12 20
```

样例输出 1

```
4 4 5 5 5 6 6 6 6
```

样例解释 1

本样例与题目描述中的情况相同。

样例输入 2

```
9
7 4 3 5 4 6 9 3 2
32 45
```

样例输出 2

```
14 15 15 15 15 16 16 18 18 18 18 21 21 24
```

样例 3

见下发文件中的 `rectangle/rectangle3.in` 和 `rectangle/rectangle3.ans`。

数据规模与约定

本题采用子任务捆绑测试。对于每个子任务，你只有通过了这个子任务的所有数据，才能获得这个子任务的分数。

- 子任务 1（20 分）：$n \le 1000$;
- 子任务 2（15 分）：$n \le 5 \times 10^4$;
- 子任务 3（10 分）：$h_i = 1$;

- 子任务 4（20 分）：$h_i = i$；
- 子任务 5（20 分）：$R - L + 1 \le 3$；
- 子任务 6（15 分）：无特殊限制。

对于所有数据，$1 \le n \le 3 \times 10^5$，$1 \le h_i \le 10^9$，$1 \le L \le R \le \frac{n(n+1)}{2}$，$R - L + 1 \le 3 \times 10^5$。

D. 排列（permutation）

题目描述

有一个长度为 $2n$ 的数组，你可以将其划分成 k 段，使得每段长度都是偶数（不能为空）。

有一个随机数生成器，每次会返回一个 1 到 $2n$ 的排列（下标从 1 开始），然后下标为偶数的位置里的数会被拿出来，排序然后放回去。

然后生成的排列按顺序放到数组中，**一个划分的价值为每段逆序对个数乘积的期望。**

比如产生排列 $\{6,4,1,2,5,3\}$ 会变成 $\{6,2,1,3,5,4\}$，假设这个数组被划成了 $[1,4],[5,6]$，那么第一段内逆序对个数为 4，第二段内逆序对个数为 1，所以乘积为 4。

求**所有合法划分的价值总和**，对一个质数 P 取模。比如将长度为 6 的数组分成 2 段，只有可能是 $[1,4],[5,6]$ 和 $[1,2],[3,6]$。

输入格式

从文件 `permutation.in` 中读入数据。

本题的每个测试点包含多组询问。

第一行包含一个质数 P 表示模数。

第二行包含一个整数，表示询问组数。

接下来 T 行，每行两个整数表示 n,k。

输出格式

输出到文件 `permutation.out` 中。

对于每组询问，输出一行一个整数表示答案在模 P 意义下的值，共输出 T 行。

可以证明，答案均能表示为既约分数 $\frac{p}{q}$ 的形式，你只需要输出满足 $qx \equiv p \pmod{P}$ 且 $0 \le x < P$ 的整数 x。可以证明，这样的整数 x 是唯一的。

样例

样例输入

```
3246307
6
1 1
2 1
2 2
3 1
3 2
3 3
```

样例输出

```
1623154
2705258
811577
5
1298525
2028942
```

数据规模与约定

本题共有 10 个测试点，每个测试点 10 分。

对于 10% 的数据，$n \le 5$，$P = 10^9 + 7$；
对于 20% 的数据，$n \le 10$，$P = 10^9 + 7$；
对于 50% 的数据，$n \le 40$；
另有 20% 的数据，$k = 1$；
另有 20% 的数据，$k = n$；
对于 100% 的数据，$1 \le k \le n \le 100$，$5000 \le P \le 10^9 + 7$，$1 \le T \le 10^4$。