

NO | P模拟赛

题目	英文题目名	时间限制	空间限制
过河	river	1 s	128 MB
距离	dis	1 s	128 MB
维护数据库	database	1 s	256 MB
快速排序	sort	1 s	256 MB

过河

题目描述

前面有一条河，有 n 个人想要过河，岸的这边有一条船。

操作一条船最少需要 L 个人，而船最多能载 R 个人。

划船是个非常令人疲惫的事情，每次用这艘船运送一群人到河的另一边时，船上所有的人都必须有大于0的精力值，并且到达后船上所有人的精力值会减少 1。

最初，第 i 个人的精力值为 h_i 。

你需要判断用这艘船是否可以把所有人运到对岸。

输入格式

第一行一个整数 T 表示数据组数。

对于每组数据，

第一行输入三个整数 n, L, R ($1 \leq L < R \leq n \leq 5 * 10^5$)，分别表示人的数量，每次使用一条船的最少和最多的人数。

第二行输入 n 个整数 h_1, h_2, \dots, h_n ($1 \leq h_i \leq 5 * 10^5$)， h_i ($1 \leq i \leq n$) 表示第 i 个人的初始精力值。

输出格式

对于每组数据，如果可以用这艘船将所有人运到对岸，输出 "Yes"（不含引号），否则输出 "No"（不含引号）

样例 #1

样例输入 #1

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

样例输出 #1

```
Yes
```

样例 #2

样例输入 #2

1
5 1 2
1 1 1 1 5

样例输出 #2

No

样例 #3

见下发文件。

数据范围

对于 20% 的数据，保证 $n \leq 10$

对于 50% 的数据，保证 $n \leq 2000$

对于另外 20% 的数据，保证 $L = n - 2, R = n - 1$

对于 100% 的数据，保证

$$1 \leq L < R \leq n \leq 5 * 10^5, 1 \leq h_i \leq 5 * 10^5, T \leq 10, \sum n \leq 10^6$$

距离

题目描述

闪电麦昆正在和哈德逊医生在温泉小镇进行训练，温泉小镇可以看成是一张 n 个点 m 条边的无向图。

为了拿下下一届活塞杯冠军，哈德逊决定每天给闪电麦昆指定一条简单路径作为练习赛道。由于下届比赛群贤毕至，少长咸集，竞争压力非常大，练习时赛道显然是越长越好。又由于温泉小镇某些地方地形非常特殊，特别考验赛车选手的某些技巧，于是哈德逊医生想找经过某个节点的赛道。作为前活塞杯冠军，哈德逊大黄蜂当然知道怎么对闪电麦昆进行训练，但是在指定路径的过程中，哈德逊犯了难。

哈德逊医生想知道，对于每一个节点，经过他的最长的简单路径长度为多少。而你作为温泉小镇唯一的编程选手，哈德逊医生将这个问题交给你。

注：简单路径是指图中路径上的顶点都不相同的路径。

输入格式

第一行两个整数 n, m ，表示图的点数和边数。

接下来 m 行，每行两个整数 x, y ，表示节点 x 和 y 间有边相连。

保证图是一张连通图，且无重边无自环。

输出格式

一行 n 个整数，第 i 个整数表示经过第 i 个点的一条最长的简单路径的长度。

样例 #1

样例输入 #1

```
8 7
1 2
2 3
2 4
4 6
5 4
5 7
5 8
```

样例输出 #1

```
4 4 4 4 4 3 4 4
```

样例 #2

样例输入 #2

```
8 8
1 2
2 3
2 4
4 6
5 4
5 7
5 8
3 6
```

样例输出 #2

```
6 6 6 6 6 6 6 6
```

样例 #3

见下发文件。

数据范围

测试点编号	$n \leq$	$m =$
1 ~ 2	2×10^3	$n - 1$
3 ~ 6	2×10^3	n
7 ~ 12	10^5	$n - 1$
13 ~ 20	10^5	n

维护数据库

题目描述

头还我爷爷为了训练七娃的编程能力，于是让大娃、二娃、三娃、四娃、五娃和六娃给七娃出一道数据库练习题。

六娃说，我给你一个初始包含 n 个整数的数据库，即 a_1, a_2, \dots, a_n 。然后大哥、二哥、三哥、四哥和五哥会对数据库进行 m 次操作。操作分别为：

- 1 x ，大娃会向数据库里加入一个整数 x 。
- 2 x ，二娃会将数据库中每个整数或上 x 。
- 3 x ，三娃会将数据库中每个整数与上 x 。
- 4 x ，四娃会将数据库中每个整数异或上 x 。
- 5 x ，五娃会询问数据库中每个整数异或上 x 的最大值。

七娃并不会这道题，于是他用葫芦抓住了你，你现在需要帮他解决这一难题。

输入格式

第一行两个整数 n, m ，表示初始整数数目和操作次数。

第二行 n 个整数，表示初始的每个整数。

接下来 m 行，每行两个整数 opt, x ，表示操作种类与参数 x 。

输出格式

对于每个 5 类型操作，输出一行一个整数，表示答案。

样例 #1

样例输入 #1

```
7 10
1 9 1 9 8 1 0
1 1
4 5
1 4
5 6
4 5
5 3
3 2
5 1
2 9
5 7
```

样例输出 #1

11
11
1
14

样例 #2

见下发文件。

数据范围

测试点编号	$n \leq$	$m \leq$	特殊性质
1	2×10^3	2×10^3	无
2	10^5	10^5	$opt \in \{1, 5\}$
3	10^5	10^5	$opt \in \{4, 5\}$
4	10^5	10^5	$opt \in \{1, 4, 5\}$
5 ~ 7	10^5	10^5	$opt \in \{2, 3, 5\}$
8 ~ 10	10^5	10^5	无

对于 100% 的数据， $1 \leq n, m \leq 10^5$ ， $1 \leq opt \leq 5$ ， $0 \leq a_i, x < 2^{31}$ 。

快速排序

题目描述

请注意，本题中部分名词含义和常见含义有所区别，请注意分辨。

定义一次冒泡排序为：

```
void bubble_sort(a) {
    for( int i = 0; i < a.size() - 1; i ++ )
        if( a[i] > a[ i + 1 ] )
            std::swap( a[i], a[ i + 1 ] );
}
```

即，从该数组的第一个位置开始，设当前进行到的位置为 i ，若 $A_i > A_{i+1}$ 则交换。

定义一次快速排序为：

```
void quick_sort(a) {
    if( a.size() == 1 )
        return ;
    while( a has no partition points ) {
        cnt += a.size()
        bubble_sort(a)
    }
    divide a at all partition points;
    do quick_sort for each_part
}
```

即，对于当前递归到的区间 $[l, r]$ ，定义一个位置 i 为分割点，当且仅当 $\forall x \in [l, i], y \in [i + 1, r], A_x < A_y$ 。

快速排序的过程是这样的，对于当前的区间 $[l, r]$ ，若该区间长度为 1，直接返回；

否则，若该区间中存在分割点，则找出所有分割点，递归两两相邻分割点中的区间。

即，若当前递归区间为 $[l, r]$ ，分割点为 $p_1, p_2, \dots, p_k, \forall p_i \in [l, r]$ 则递归 $[l, p_1], [p_1 + 1, p_2], \dots, [p_{k-1} + 1, p_k], [p_k + 1, r]$ 。

若不存在分割点，则执行多次冒泡，每一次冒泡将 cnt 的值加上当前数组长度，直至出现分割点，执行递归。

给出两个正整数 L, R ，对所有的 $n \in [L, R]$ 分别求出所有长度为 n 的排列的 cnt 值之和对 998244353 取模的结果。

输入格式

一行两个正整数 L, R ，意义同题目描述。

输出格式

为减少输出量，仅输出一个整数表示所有答案的异或和。

样例 #1

样例输入 #1

2 8

样例输出 #1

920329

样例 #2

样例输入 #2

17 26

样例输出 #2

206664280

样例 #3

样例输入 #3

60103 201518

样例输出 #3

919666446

数据范围与提示

测试点编号	$R - L + 1$	R
1 ~ 2	≤ 10	≤ 8
3 ~ 4	≤ 10	≤ 20
5 ~ 6	≤ 10	≤ 500
7 ~ 8	≤ 5000	≤ 5000
9 ~ 10	$\leq 10^7$	$\leq 10^7$