

# 必可赛前公益众筹赛

## 第九试

时间：2024 年 11 月 2 日 08:00 ~ 12:00

题目名称	序列划分	二进制的世界	长者	种植计划
题目类型	传统型	传统型	传统型	传统型
目录	dos	binary	young	tree
可执行文件名	dos	binary	young	tree
输入文件名	dos.in	binary.in	young.in	tree.in
输出文件名	dos.out	binary.out	young.out	tree.in
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	128 MB	2014 MB
测试点数目	10	20	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	dos.cpp	binary.cpp	young.cpp	tree.cpp
-----------	---------	------------	-----------	----------

编译选项

对于 C++ 语言	-lm -O2 -std=c++14
-----------	--------------------

### 注意事项与提醒（请选手务必仔细阅读）

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立好的，且带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
4. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
5. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
6. 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**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
13. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
14. 请务必使用题面中规定的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。
15. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**

## 【前言】

弄一套合适的题目是很费时间和精力事情，这是我们第一次尝试公益活动，活动的初衷是为自觉比赛的选手提供一些比赛选项「能够离线完成最佳」，想把我们训练中遇到的较为合适或者很有启发性的题目尽量展示给大家，这个过程不可避免地遭遇原题，事实上，任何一个能够稳定输出的学校，都是依据大量的、成熟的原题进行训练。这是无可厚非的。

哪怕是组一套试题，其实也具有很大的工作量，部分分的配置、标程、数据、交互库或 SPJ、题解、题面等等。

希望大家，认真对待每一道题目，即使知道来源，请君勿多言，自己做自己的就行了，在市面价 200 ~ 500 一场的现状下，包含退费和奖励的 9.9 元报名费实在是微不足道的。

最后，我们欣赏行胜于言的作风，通过行动来改变，能够提供哪怕一道题目也远胜喋喋不休的抱怨。

## 序列划分 (dos)

### 【题目描述】

给定  $n$  和  $m$ ，问是否能把 1 到  $n$  这  $n$  个数划分到  $m$  个集合中，使得每个集合的元素之和相等。

如果能，输出任意一种方案即可。

### 【输入格式】

从文件 *dos.in* 中读入数据。

输入一行，包含一个整数  $t$ ，表示数据组数。

接下来  $t$  行，每行两个整数  $n, m$ ，含义如上所述。

### 【输出格式】

输出到文件 *dos.out* 中。

对于每组数据，如果能做这样的划分，第一行先输出 YES，接下来  $m$  行，每行先输出一个整数  $t$ ，表示这个集合有  $t$  个元素，然后再输出  $t$  个整数，表示该集合中的元素。

集合的输出顺序和某个集合中元素的顺序可任意输出。

如果不能做这样的划分，只输出一行 NO。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 NO
2 YES
3 1 5
4 2 1 4
5 2 2 3
6 NO
7 YES
8 3 1 5 9
9 3 2 6 7
```

10 | 3 3 4 8

**【数据范围】**

对于 20% 的数据,  $n \leq 20$ ;

对于另外 10% 的数据,  $m = 2$ ;

对于另外 20% 的数据,  $m = 4$ ;

对于所有数据,  $1 \leq \sum n \leq 2 \times 10^6, 1 \leq n \leq 10^5, 2 \leq m \leq 10$ 。

## 二进制的世界 (binary)

### 【题目描述】

在二进制的世界里，只有三种运算：按位与 (and)、按位或 (or)、按位异或 (xor)。

在这个世界里，有  $n$  个人从左往右排成一排，其中从左往右第  $i$  个人手上写着一个数字  $a_i$ 。

每个人都想在他左边找到他的好朋友，第  $i$  个人和第  $j$  个人之间的友好程度为  $a_i \text{ opt } a_j$ ，其中 opt 是三种运算中的一种。

请对于从左往右第 2 到第  $n$  个人，求出他左边和他友好程度最大的人，以及达到最大值的人数。

### 【输入格式】

从文件 *binary.in* 中读入数据。

第一行包含一个正整数  $n$ ，一个串 opt 和一个整数 type，分别表示人数，运算类型以及数据类型。

第二行包含  $n$  个整数，其中第  $i$  个数为  $a_i$ ，表示从左往右第  $i$  个人手上的数字。

### 【输出格式】

输出到文件 *binary.out* 中。

如果 type = 0，输出  $n - 1$  行，每行一个整数，其中第  $i$  行输出和第  $i + 1$  个人的友好程度的最大值。

如果 type = 1，输出  $n - 1$  行，每行两个整数，其中第  $i$  行先输出和第  $i + 1$  个人的友好程度的最大值，再输出达到最大值的人数。

### 【样例 1 输入】

```
1 5 and 1
2 3 5 2 7 1
```

### 【样例 1 输出】

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

### 【数据范围】

对于 100% 的数据， $0 \leq a_i \leq 65535$ 。

测试点编号	$n =$	opt	type
1	1000	and	1
2	1000	or	1
3 ~ 4	100000	xor	1
5	100000	and	0
6	100000	or	0
7 ~ 8	100000	and	1
9 ~ 10	100000	or	1

## 长者 (young)

### 【题目描述】

西方有  $n$  个国家，长者决定向西方的每个国家普及人生经验，但首先要让他们互通火车，第  $i$  个国家有一个权值  $A_i$ ，修建连接第  $i$  个国家到第  $j$  个国家的铁路，需要付出  $A_i \text{ xor } A_j$  (xor 表示按位异或) 的代价，长者希望代价总和尽量小(也就是选择一个最小生成树)。

但是在长者以前，没人去过西方，所以不知道每个国家的权值。但是我们知道每个国家的权值都是一个在 0 到  $2^m - 1$  之间的随机整数，长者希望知道他所需要付出的代价的期望。

当然，答案是一个有理分数，为了避免精度误差，长者需要你输出这个分数在模 258280327 ( $2 \times 3^{17} + 1$ ，一个质数) 意义下的值 (如果不存在则输出  $-1$ )。

### 【输入格式】

从文件 *young.in* 中读入数据。

一行两个正整数，分别表示  $n, m$ 。

### 【输出格式】

输出到文件 *young.out* 中。

一行一个正整数表示答案。

### 【样例 1 输入】

1 2 2

### 【样例 1 输出】

1 129140165

### 【样例 2 输入】

1 10 3

### 【样例 2 输出】

1 7008635

### 【数据范围】

对于 30% 的数据,  $n \times m \leq 16$ ;

对于另外 20% 的数据,  $m \leq 4$ ;

对于 100% 的数据,  $n \leq 50, m \leq 8$ ;



## 种植计划 (tree)

### 【题目描述】

小 Z 种了一排树。这些树共有  $n$  棵，等间距的排在一条路上，第  $i$  棵树的初始高度为  $h_i$ 。

小 Z 希望这些树苗茁壮成长。因此，她有时会调查第  $l$  棵到第  $r$  棵树的高度之和。同时，她还会给这些树施肥。当她给第  $l$  棵到第  $r$  棵的每一棵树施用品质为  $c$  的肥料时，这些树的高度会立刻分别增加  $c$ 。当然，小 Z 有可能买到盗版货，因此  $c$  可能小于 0。这时候，这些树的高度会立刻减少  $c$ 。

小 Z 养了一只猫，这只猫喜欢在树上蹦跳。为了避免小猫有危险，小 Z 有时会对一段树木进行修剪。当小 Z 修剪第  $l$  棵到第  $r$  棵树时，她可以从这段区间中选择若干棵树使它们的高度减小，使得相邻两棵树的高度差不大于  $k$ ，且树的高度和尽可能大。

形式化的，假设当前第  $l$  棵到第  $r$  棵树的高度分别为  $a_l, a_{l+1}, \dots, a_r$ ，小 Z 会将它们变为  $b_l, b_{l+1}, \dots, b_r$ ，并遵循以下原则：

- $\forall l \leq i \leq r$ ，使得  $b_i \leq a_i$ 。
- $\forall l \leq i < r$ ，使得  $|b_i - b_{i+1}| \leq k$ 。
- 在满足上述条件的同时，使得  $\sum_{i=l}^r b_i$  尽可能大。

可以证明，满足上述条件的  $b_l, b_{l+1}, \dots, b_r$  是唯一的。

请帮助小 Z 完成这几种操作。

值得注意的是，小 Z 生活在一个魔幻的世界中，因此树木的高度可以是负数或 0。

### 【输入格式】

从文件 `tree.in` 中读入数据。

第一行，三个非负整数  $n, m, k$ 。

第二行， $n$  个整数  $h_1, h_2, \dots, h_n$ 。

接下来  $m$  行，每行首先有一个整数  $op$ ，代表操作类型。

- $op = 1$ ，则再输入两个整数  $l, r$ ，表示小 Z 调查第  $l$  棵到第  $r$  棵树的高度之和。
- $op = 2$ ，则再输入三个整数  $l, r, c$ ，表示小 Z 给第  $l$  棵到第  $r$  棵树施用品质为  $c$  的肥料。
- $op = 3$ ，则再输入两个整数  $l, r$ ，表示小 Z 对第  $l$  棵到第  $r$  棵树进行修剪。修剪的定义见题目描述。

### 【输出格式】

输出到文件 `tree.out` 中。

对于每一个  $op = 1$  的操作，输出一行，表示答案。

**【样例 1 输入】**

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

**【样例 1 输出】**

```
1 15
2 11
3 11
```

**【样例 1 解释】**

序列初始为 2, 4, 5, 3, 1, 执行「2 2 4 1」后为 2, 5, 6, 4, 1, 执行「3 1 3」后为 2, 3, 4, 4, 1。

**【样例 2】**

见选手目录下的 `tree/ex_tree2.in` 与 `tree/ex_tree2.ans`。

**【数据范围】**

对于前 10% 的数据,  $n, m \leq 5, h_i \leq 5, k \leq 2, -2 \leq c \leq 2$ 。

对于前 30% 的数据,  $n, m \leq 10^3$ 。

对于另外 10% 的数据,  $op = 3$  的操作不超过 10 次。

对于另外 10% 的数据,  $op = 3$  的操作不超过 100 次。

对于另外 30% 的数据, 在第  $i$  次操作中,  $op \equiv i \pmod{3}$ , 且当  $op = 3$  的时候,  $l = 1, r = n$ 。

对于 100% 的数据,  $1 \leq n, m \leq 10^5, 1 \leq h_i \leq 10^9, 1 \leq l \leq r \leq n, 1 \leq k \leq 10^4, -10^4 \leq c \leq 10^4$ 。