

3 位运算

3.1 Problem Statement

小 D 正在研究位运算。

众所周知，常见的二元位运算有如下三种：

1. **按位与**（即 C++ 中的 `&`），记作 $x \text{ and } y$ ；
2. **按位异或**（即 C++ 中的 `^`），记作 $x \text{ xor } y$ ；
3. **按位或**（即 C++ 中的 `|`），记作 $x \text{ or } y$ ；

小 D 想了一个长度为 n 的序列 a_1, a_2, \dots, a_n 和一个位运算符 op 。

小 D 想要知道，如果他在序列中任取两个元素 a_i, a_j ，并对他们使用运算符 op ，可能得到的最大结果。形式化地，小 D 想要计算下式的值：

$$\max_{1 \leq i < j \leq n} \{a_i \text{ op } a_j\} \quad (1)$$

此外，小 D 还想要知道，有多少对 (i, j) 可以使得上式取到最大值。

但是小 D 并不会，请你帮帮他。

3.2 Input Format

从标准输入读入数据。

第一行两个正整数 n, q ，其中 n 表示序列长度， q 表示操作符类型：

- 若 $q = 1$ ，则操作符 op 为**按位与**运算（and）；
- 若 $q = 2$ ，则操作符 op 为**按位异或**运算（xor）；
- 若 $q = 3$ ，则操作符 op 为**按位或**运算（or）；

第二行 n 个整数 a_1, a_2, \dots, a_n ，表示序列中的元素。

3.3 Output Format

向标准输出输出答案。

输出一行两个整数，表示小 D 要求出的最大值以及取到最大值的 (i, j) 个数。

3.4 Sample 1

3.4.1 Input

```
5 1
1 4 5 7 9
```

3.4.2 Output

5 1

3.5 Sample 2

3.5.1 Input

5 2
2 3 4 5 7

3.5.2 Output

7 2

3.6 Sample 3

3.6.1 Input

5 3
1 2 4 5 9

3.6.2 Output

13 2

3.7 Sample Explanations

- 对于第一组样例，唯一的最大值 5 在 $(i, j) = (3, 4)$ 时取到。
- 对于第二组样例，最大值 7 在 $(i, j) = (1, 4)$ 或 $(2, 3)$ 时取到。
- 对于第三组样例，最大值 13 在 $(i, j) = (3, 5)$ 或 $(4, 5)$ 时取到。

3.8 Constraints

对于所有测试数据， $2 \leq n \leq 10^5$ ， $0 \leq a_i < 2^{23}$ ， $q \in \{1, 2, 3\}$ 。

- 子任务 1 (20 分)： $n \leq 1000$ 。
- 子任务 2 (25 分)： $a_i < 8192$ 。
- 子任务 3 (15 分)： $q = 1$ 。
- 子任务 4 (15 分)： $q = 2$ 。
- 子任务 5 (15 分)： $q = 3$ 。
- 子任务 6 (10 分)： 无特殊限制。