

Xor

Cho dãy số $a = (a_1, a_2, \dots, a_n)$. Bạn cần biến dãy số a thành dãy số $b = (b_1, b_2, \dots, b_n)$ bằng cách thực hiện một số (có thể 0 hoặc nhiều hơn) các thao tác cho phép sau đây. Các thao tác hợp lệ được mô tả bởi dãy số $x = (x_1, x_2, \dots, x_m)$. Ở mỗi thao tác bạn được phép chọn phần tử x_j bất kỳ ($1 \leq j \leq m$), sau đó với mỗi phần tử a_i ($1 \leq i \leq n$) bạn hoặc giữ nguyên nó hoặc thay đổi giá trị của nó thành $a_i \text{ XOR } x_j$. Chi phí của thao tác này là 2^j .

Yêu cầu: Hãy xác định tổng chi phí nhỏ nhất để có thể chuyển dãy a thành dãy b .

Input: đọc từ file **xor.in**

Dòng đầu tiên chứa số nguyên T ($1 \leq T \leq 10$) là số lượng test. T nhóm dòng sau, mỗi nhóm dòng mô tả một test. Mỗi test có định dạng:

- Dòng đầu tiên chứa số nguyên dương n .
- Dòng thứ hai chứa n số nguyên không âm a_1, a_2, \dots, a_n .
- Dòng thứ ba chứa n số nguyên không âm b_1, b_2, \dots, b_n .
- Dòng tiếp theo chứa số nguyên dương m .
- Dòng cuối cùng chứa m số nguyên không âm x_1, x_2, \dots, x_m .

Output: ghi ra file **xor.out**

Với mỗi test, theo đúng thứ tự được cho trong input, in ra trên một dòng chi phí nhỏ nhất để chuyển dãy số a thành dãy số b , hoặc -1 nếu không chuyển được. Do chi phí nhỏ nhất có thể rất lớn, bạn chỉ cần in phần dư của nó sau khi chia cho $10^9 + 7$.

Subtask:

Subtask 1 (20%): $n, a_i, b_i, m, x_i < 16$

Subtask 2 (30%): $n, a_i, b_i, m, x_i < 64$

Subtask 3 (30%): $n, a_i, b_i, m, x_i < 256$

Subtask 4 (20%): $n, a_i, b_i, m, x_i < 1024$.

Ví dụ

xor.in	xor.out	Giải thích
2 3 0 2 3 3 0 2 3 1 2 3 1 1 3 1 1	6 -1	Test #1: một cách chuyển tối ưu là: Thao tác 1: chọn $x_1 = 1$, $a = (0, 2, 3) \rightarrow (0 \text{ XOR } 1, 2, 3 \text{ XOR } 1) = (1, 2, 2)$ Thao tác 2: chọn $x_2 = 2$, $a = (1, 2, 2) \rightarrow (1 \text{ XOR } 2, 2 \text{ XOR } 2, 2) = (3, 0, 2)$ Tổng chi phí là $2^1 + 2^2 = 6$ Test #2: không thể chuyển 1 về 3 được

