

Задача D. Xoractive

Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Айдос придумал новую головоломку для Темирулана. Он загадал последовательность a из n неотрицательных целых чисел, пронумерованных от 1 до n : a_1, a_2, \dots, a_n .

Темирулан может задавать один из двух типов вопросов:

- ask — Узнать i -е число в загаданной последовательности.
- $get_pairwise_xor$ — Для заданной последовательности различных чисел: i_1, i_2, \dots, i_k , узнать множество попарных значений xor для элементов последовательности a в индексах i_1, i_2, \dots, i_k , $\{a_{i_x} \oplus a_{i_y} \mid 1 \leq x, y \leq k\}$.

Например, если Айдос загадал последовательность $[1, 5, 6, 3]$. Тогда на вопрос $ask(2)$ Айдос ответит числом 5, а на вопрос $get_pairwise_xor(\{3, 4\})$ Айдос ответит последовательностью $[0, 0, 5, 5]$, так как

- $a_3 \oplus a_4 = 6 \oplus 3 = 5$
- $a_4 \oplus a_3 = 3 \oplus 6 = 5$
- $a_3 \oplus a_3 = 6 \oplus 6 = 0$
- $a_4 \oplus a_4 = 3 \oplus 3 = 0$.

Темирулан не справился с этой головоломкой и вам требуется помочь Темирулану. Отгадайте последовательность Айдоса, используя вышеописанные вопросы.

Протокол взаимодействия

ВАША ПОСЫЛКА НЕ ДОЛЖНА НИКАКИМ ОБРАЗОМ ВЗАИМОДЕЙСТВОВАТЬ СО СТАНДАРТНЫМИ ПОТОКАМИ ВВОДА И ВЫВОДА, А ТАКЖЕ С ЛЮБЫМИ ДРУГИМИ ФАЙЛАМИ.

Вам надо реализовать одну функцию: `int [] guess(int n)`

- n : длина загаданной последовательности.
- Функция вызывается ровно один раз для каждого теста.
- Функция должна вернуть загаданную последовательность в том же порядке.

Ваша программа может вызывать следующие функции:

1. `int ask(int i)`

- i : индекс в загаданной последовательности, $1 \leq i \leq n$.
- Функция возвращает i -й элемент в загаданной последовательности.

2. `int [] get_pairwise_xor(int [] pos)`

- pos : **непустой** список индексов последовательности.
- Все элементы pos должны быть **различными** целыми числами.
- Пусть k это количество элементов в pos . Тогда $1 \leq pos_i \leq n$ для всех $1 \leq i \leq k$.
- Функция возвращает отсортированный список из k^2 элементов: множество попарных значений xor , $\{a_{pos_x} \oplus a_{pos_y} \mid 1 \leq x, y \leq k\}$.

Суммарно вы можете вызывать обе функции не более 200 раз для каждого теста. Если какие-либо из вышеописанных условий не выполняются, ваша программа получит вердикт **Wrong Answer**. В противном случае, ваша программа получит вердикт **Accepted**, и ваш балл определяется суммарным количеством вызовов функции ask и $get_pairwise_xor$ (Смотрите раздел “Система оценки”).

Система оценки

- $2 \leq n \leq 100$
- $0 \leq a_i \leq 10^9$, для всех $1 \leq i \leq n$.

В этой задаче проверяющий модуль НЕ ЯВЛЯЕТСЯ адаптивным. Это означает, что последовательность a зафиксирована на момент запуска проверяющего модуля и не зависит от выполненных запросов в вашем решении.

1. (6 баллов) $n \leq 4$
2. (94 балла) Без дополнительных ограничений. Для этой подзадачи ваш результат за каждый тест вычисляется следующим образом. Пусть суммарное количество вызовов функции ask и $get_pairwise_xor$ равно q .
 - Если $q \leq 15$, ваш результат 94.
 - Если $15 < q \leq 40$, ваш результат $84 - 2(q - 16)$.
 - Если $40 < q \leq 50$, ваш результат 35.
 - Иначе, ваш результат 0.

Обратите внимание, что результат за каждую подзадачу равен минимуму среди результатов на тестах этой подзадачи.

Замечание

Под операцией xor подразумевается побитовое исключающее ИЛИ.

Пусть массив a равен $[1, 5, 6, 3]$. Проверяющий модуль (grader) вызывает функции. Пример взаимодействия ниже.

Вызов	Результат
$ask(2)$	5
$get_pairwise_xor(\{1, 2, 3\})$	$\{0, 0, 0, 3, 3, 4, 4, 7, 7\}$
$ask(3)$	6
$get_pairwise_xor(\{4, 2\})$	$\{0, 0, 6, 6\}$
$get_pairwise_xor(\{2\})$	$\{0\}$

Пример проверяющего модуля читает входные данные в следующем формате:

- Строка 1: n
- Строка 2: a_1, a_2, \dots, a_n

В системе ВЫ МОЖЕТЕ СКАЧАТЬ `xoractive.zip`, где есть примеры по языкам Java, C++11, FPC, Python 2.

Все вызовы функций можно увидеть выше. Отличием является язык Python 2. Вам надо будет реализовать функцию `def guess(n, interactor)`, `interactor` это instance тестирующего класса, функции `ask` и `get_pairwise_xor` являются методами данного класса.

`xoractive.zip` — содержит примеры решений по каждому языку.

В решении на языке Java файл и имя класса должны называться `Xoractive.java` и `Xoractive` соответственно.

В решении на языке Pascal файл должен называться `xoractive.pas`.

Задача Е. Отрезок подлиннее

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Наш маленький мальчик Асхат сделал интересное наблюдение — покрыть массив “прыжками,” где каждый “больше” предыдущего, не так уж и просто. Разумеется, теперь Вам предстоит придумать как это сделать.

Вам дается последовательность из положительных целых чисел длины N .

Нужно разделить данную последовательность на максимальное количество отрезков так, что:

1. Каждый элемент последовательности принадлежит ровно одному отрезку.
2. Сумма чисел в каждом отрезке, кроме первого, не меньше чем в предыдущем.

Формат входных данных

Первая строка ввода содержит целое число N ($1 \leq N \leq 5 \cdot 10^5$).

Следующая строка содержит N положительных целых чисел a_i ($1 \leq a_i \leq 10^9$) разделенных пробелами.

Формат выходных данных

Выведите единственное целое число — максимальное число отрезков, на которое можно поделить данную последовательность.

Система оценки

Данная задача содержит пять подзадач, в каждой подзадаче выполняются дополнительные ограничения:

1. $1 \leq N \leq 20$, $a_i \leq 10^6$. Оценивается в 13 баллов.
2. $1 \leq N \leq 500$. Оценивается в 14 баллов.
3. $1 \leq N \leq 3000$. Оценивается в 10 баллов.
4. $1 \leq N \leq 10^5$. Оценивается в 36 баллов.
5. Ограничения из условия. Оценивается в 27 баллов.

Примеры

стандартный ввод	стандартный вывод
4 2 3 1 7	3
5 6 2 3 9 13	3
3 3 1 2	2

Задача F. Код "Любойн"

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Команда "Любойн" - Асхат, Санжар и Нурбакыт решили, что им пора немного расширить свой кругозор и принялись изучать электронику. Плодом их упорных трудов стало их новое изобретение: "Коммутатор Любойн" - устройство, преобразующее получаемый на вход аналоговый сигнал в своеобразной манере.

Сигнал можно представить как последовательность из N битов. Выходной сигнал коммутатора отличается от входного в ровно K битах, и никакие два сигнала поданных на вход не произведут одинаковый выходной сигнал. Дабы сделать их изобретение еще более дерзким, ребята добавили особую "фичу": если бинарный параметр T выставлен равным 1, связанная последовательность выходных сигналов зациклена, то есть если начать с произвольного сигнала, "скормить" его коммутатору и заменить его выходным сигналом, а затем повторить описанную процедуру достаточно много раз, то в какой-то момент мы обязательно получим изначальный сигнал. Это, однако, не выполняется, если параметр T выставлен равным 0.

В данной задаче Вам требуется повторить пока непобитое достижение команды: воссоздать код "Любойн" - отображение входных сигналов к выходным заложенное в коммутатор. Чтобы упростить вашу задачу, Вам потребуется вывести лишь связную последовательность выходных сигналов, как описано выше, начиная с сигнала S .

Более формально, Вам нужно найти последовательность f длины 2^N , состоящую из двоичных чисел длины N (включая лидирующие нули), такую, что:

1. $f_0 = S$
2. Для любых i и j ($i \neq j$), $f_i \neq f_j$
3. Для любых i ($0 \leq i < 2^N - 1$), f_i отличается от f_{i+1} в ровно K цифрах в двоичной записи. К тому же, если параметр T равняется 1, то последовательность должна быть зациклена, то есть значение f_{2^N-1} должно также различаться от f_0 в ровно K цифрах в двоичной записи.

Формат входных данных

Первое строка ввода содержит три целых числа N , K , и T ($2 \leq N \leq 18$, $1 \leq K < N$, $0 \leq T \leq 1$).
Вторая строка ввода содержит двоичную запись начального числа S .

Формат выходных данных

Если указанной последовательности не существует, выведите -1.

Иначе, первая строка вывода должна содержать размер искомой последовательности — 2^N . Строки со 2 по $2^N + 2$ должны содержать по одному двоичному числу — значение f_{i-2} .

Если существует более одного правильного решения, выведите любое из них.

Система оценки

Данная задача содержит восемь подзадач:

1. Тест из условия. Оценивается в 0 баллов.
2. $N = 4, K = 3, T = 1, S = 0$. Оценивается в 5 баллов.
3. $2 \leq N \leq 18, K$ — четное, $T \leq 1, S < 2^N$. Оценивается в 3 баллов.
4. $2 \leq N \leq 18, K = 1, T = 1, S = 0$. Оценивается в 11 баллов.
5. $2 \leq N \leq 18, K = 3, T = 0, S = 0$. Оценивается в 15 баллов.
6. $2 \leq N \leq 18, K \cdot 2 < N, T = 0, S < 2^N$. Оценивается в 18 баллов.
7. $2 \leq N \leq 18, K < N, T = 0, S < 2^N$. Оценивается в 31 баллов.
8. $2 \leq N \leq 18, K < N, T = 1, S < 2^N$. Ограничения из условия. Оценивается в 17 баллов.

Пример

стандартный ввод	стандартный вывод
2 1 1	4
10	10
	11
	01
	00