

Максимальный XOR

С помощью сортировки пузырьком можно получить любую перестановку блоков в обоих числах, меняя соседние. Это же верно и для итогового *XOR*. Потому что если мы смогли получить какой-то *XOR*, то можно менять в нем соседние блоки местами, просто меняя соответствующие соседние блоки в числах a и b на тех же позициях. В частности, получив какой-то *XOR*, мы всегда можем отсортировать в нём блоки по невозрастанию.

Чтобы решить задачу для $n \leq 8$, можно просто перебрать все перестановки блоков в первом числе, посчитать *XOR* со вторым, отсортировать в нем блоки по невозрастанию и выбрать из таких *XOR*-ов наибольший.

Если a состоит только из нулей, то ответом будут просто блоки b , расположенные от больших к меньшим.

Можно заметить, что в тестах, где a состоит только из блоков 00 и 01 и b только из блоков 00, 10, записи a и b никогда не пересекаются по битам, поэтому оптимально отсортировать блоки a и b по невозрастанию.

Для следующих решений предпосчитаем da_{ij} и db_{ij} — количество блоков ij , $0 \leq i, j \leq 1$ в a и в b соответственно.

Когда оба числа состоят только из блоков 00 и 01, нужно получить как можно больше блоков 01 в *XOR*. Тогда в ответе будет префикс из $\min(da_{00}, db_{01}) + \min(da_{01}, db_{10})$ блоков 01 и суффикс из нулей.

Мы знаем, что блоки в полученном *XOR* можно отсортировать по невозрастанию. Значит для оптимального решения нам нужно получить как можно больше блоков 11 и поставить их в начало *XOR*. Среди решений с максимальным количеством 11 нужно выбрать то, в котором больше всего блоков 10 и расставить их после блоков 11. Аналогично среди всех таких решений нужно выбрать то, в котором больше всего блоков 01 и поставить их после блоков 10.

Чтобы построить общее решение, будем перебирать значение блоков в порядке 11, 10, 01, 00. Пусть мы уже рассмотрели все блоки до ij . Тогда сейчас нам нужно получить как можно больше блоков ij , чтобы максимизировать ответ. Для этого переберем значение блока xy в первом числе. Тогда $xy \text{ XOR } ij$ — значение блока из второго числа, дающего нужный *XOR* в паре с xy . Будем приписывать к ответу ij , пока $da_{xy} > 0$ и $db_{xy \text{ XOR } ij} > 0$ и уменьшать da_{xy} и $db_{xy \text{ XOR } ij} > 0$ на 1.