Задача

Алиса и Боб нашли мешок с апельсинами и яблоками. Алиса взяла себе апельсин, а Боб — яблоко. Чтобы процесс разделения оставшихся фруктов был интереснее, ребята решили поиграть в игру. Они выложили в ряд карточки и на каждой записали букву ***A*** или ***B***.

Они стали убирать карточки по одной слева направо. Когда убиралась карточка с ***A***, Алиса отдавала Бобу все свои фрукты и брала из мешка такой же набор фруктов. Таким образом количество апельсинов и яблок, имеющихся у Алисы, не менялось. Если же на карточка была с ***B***, то Боб поступал аналогично.

После того, как была убрана последняя карточка, все фрукты в мешке закончились.

Известно, сколько сначала в мешке апельсинов и яблок. Требуется найти любую последовательность карточек, которыми могли играть Алиса и Боб.

Дополнительно

Входные данные

Записанные через пробел целые **X, Y ($1 \leq X, Y \leq 10^{18}$, $X * Y > 1$)** — изначальное количество апельсинов и яблок в мешке.

Выходные данные

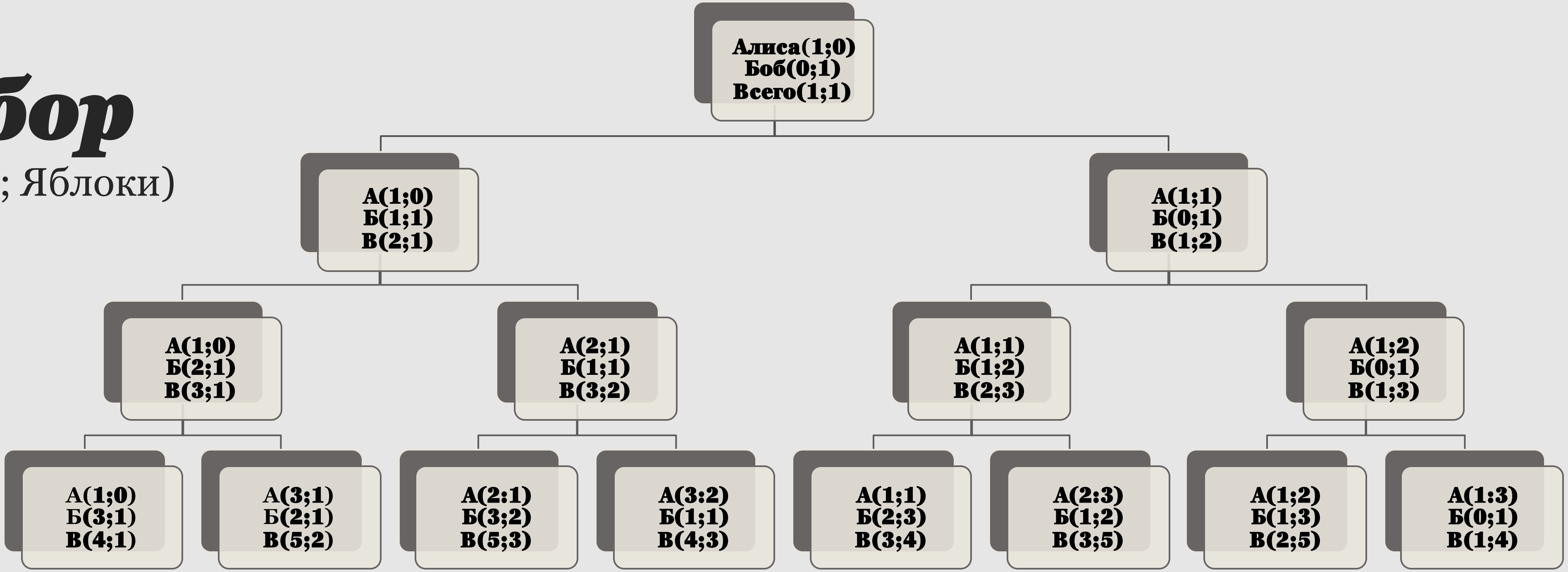
Последовательность карточек в виде строки из символов **A** и **B**, но нужно сжать строку, заменив отрезки одинаковых символов на количество повторений этого символа и его самого. Например, **AAABAABBB** нужно заменить на **3A1B2A3B**. Строка должна не превышать **10^6** символов.

Если ответ существует, то существует и его сжатое представление, не превышающее **10^6** символов. Если ответов несколько, выведите любой из них. Если не существует ответа, выведите **Impossible**.

Ограничения: 1 секунда, 256 мегабайт.

Разбор

(Апельсины; Яблоки)



В дереве описаны все возможные развития событий до 4-го хода. Если спускаться влево, то это поднятие карточки с **А**, аналогично с **В** вправо. Нас интересует количество **Всего** фруктов после каждого хода.

Видно, что с каждым спуском происходит обратный шаг алгоритма Евклида. Верхний листок дерева имеет **НОД = 1**, это значит, что и все листья имеют такой же **НОД**. Это следует из свойства: **НОД(u, v) = НОД(u - v; v)** при **$v \le u$**.

Алгоритм

Так как все листья имеют **НОД = 1**, то в мешке не могут лежать столько апельсинов и яблок, что их **НОД** был бы не равен **1**. В противном случае мы выводим **Impossible**.

Так как каждый спуск в дереве - это ход в игре, то количество ходов будет равно количеству шагов алгоритма Евклида, пока количества апельсинов и яблок не станут равными **1**.

Если в алгоритме вычитается из левого числа правое, то это Алиса передает свои фрукты. Если из правого левое, — Боб.

То есть сначала нам нужна будет проверка на **НОД = 1** с помощью обычного алгоритма Евклида, а потом запустится наша его модификация, которая будет считать подряд идущие вычитания.

Подводный камень

Вы уже знаете, что алгоритм Евклида, использующий только вычитания, серьезно проигрывает такому же, но использующему деления. (деление эквивалентно серии вычитаний, пока уменьшаемое больше вычитаемого)

Выбрав медленный аналог, мы не уложимся в **1** секунду, но, выбрав более быстрый, мы столкнемся с одной проблемой. Наши числа могут поделиться нацело, тогда одно из них станет равным нулю, а мы возвращаемся к ситуации, когда у каждого по одному фрукту.

Это может возникнуть только тогда, когда делитель был равен **1 (**потому что числа внутри пары взаимно простые). В таком случае мы будем отнимать **1** от результата деления (результат деления — это и есть количество повторений символа подряд, как вы уже сами поняли).