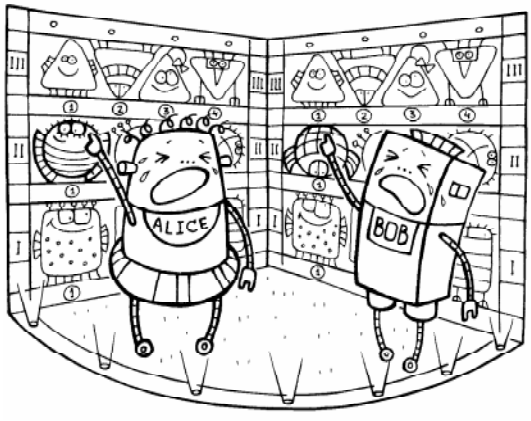
*Юрай Громкович,*
*Аблаев Фарид Мансурович*

# АЛГОРИТМЫ И СЛУЧАЙНОСТЬ УРОК 2. КОГДА СЛУЧАЙНОСТЬ МОЖЕТ БЫТЬ ПОЛЕЗНОЙ

В этой статье мы рассмотрим задачу, которая хорошо решается с применением случайности (вероятностным алгоритмом).

Что означает термин *вероятностный* алгоритм (программа)? Первые алгоритмы, разрабатываемые в курсе информатики, и конструируемые на их основе компьютерные программы являются детерминированными: такие алгоритмы и соответствующие им программы при заданных входных данных выполняют некоторые однозначно определяемые вычислительные шаги. Вероятностный алгоритм при одних и тех же входных данных может проводить вычисления разными путями, при этом каждый раз выбор конкретного пути вычисления осуществляется вероятностно. А именно, придается источник случайности, например устройство, «честно» подбрасывающее «правильную» монету, или кубик, или многогранник и т. д. Такие устройства принято называть датчиками случайных символов. Результаты бросания (символы, выданные этим датчиком) используются для выбора дальнейшего пути вычисления.




*Наша задача построить надежный и быстрый алгоритм проверки идентичности этих баз данных.*

Итак при разработке алгоритма будем считать, что в нашем распоряжении имеется необходимый нам идеальный источник случайных символов. А как поступают на практике, когда нужно на основе такого алгоритма строить компьютерную программу? Для этих нужд разрабатываются, конструируются и, конечно же, продаются датчики случайных символов. Например, при запросе «true random bits» поисковая система Google нашла 3 530 000 (за 0,27 сек.) записей на эту тему. Многочисленные фирмы-производители обещают, что их датчики обеспечат истинно случайные последовательности символов, которые можно использовать в вероятностных программах.

Итак, сформулируем задачу, для которой мы будем строить вероятностный алгоритм.

**Задача.** Имеются два компьютера, соединенные каналом связи. Назовем их Alice и Bob. Эти компьютеры оперируют с идентичными базами данных. Наша задача построить надежный и быстрый алгоритм проверки идентичности этих баз данных.

Алгоритмы, определяющие работы компьютеров в сети, называют *коммуникационными протоколами*, или просто *протоколами*.

**Важность задачи.** Задачи проверки идентичности данных достаточно часто возникают на практике: в поисковых системах, крупных банках с распределенной (по стране и миру) сетью филиалов, в страховых компаниях и т. д.

Такие базы данных постоянно изменяются, и все изменения должны происходить на всех компьютерах. Периодически нужно проверять, действительно ли базы данных на рассматриваемых компьютерах идентичны.

Размеры современных баз данных огромны и продолжают расти.

Крупнейшие мировые пользователи сейчас начинают внедрять системы хранения емкостью около 10–15 миллионов гигабайт (10–15 петабайт).

В научно-исследовательской лаборатории IBM Алмаден в штате Калифорния разрабатывается цифровое хранилище, которое будет содержать в себе порядка 200 000 жестких дисков и иметь суммарный объем в 120 миллионов гигабайт (120 петабайт). А ведь еще только в 2005 году самыми крупными в мире считались базы данных с объемом хранилища порядка 100 терабайт. База Yahoo! стала первой такой базой данных, которая преодолела рубеж в 100 терабайт.

Содержательно базы данных – это системы сложно устроенных таблиц, но, с точки зрения компьютера, – это последовательности битов.

Напомним двоичную шкалу классификация объема информации: бит – это один двоичный разряд, базовая единица измерения информации. Байт – единица информации, равная восьми битам. Увеличительные приставки кратны $1024 = 2^{10}$, то есть килобайт равен 1024 байтам, мегабайт – 1024 килобайтам или $1\,048\,576 = 2^{20}$ байтам, гигабайт – $2^{10}$ мегабайтам или $2^{30}$ байтам, терабайт – $2^{10}$ гигабайтам или $2^{40}$ байтам и петабайт – $2^{10}$ терабайтам или $2^{50}$ байтам. Пользуясь двоичной шкалой, представляют информацию операционные системы компьютеров.

Но в обыденной жизни нам удобна десятичная шкала классификации, поэтому часто говорят и пишут: килобайт – это тысяча байтов, мегабайт – это тысяча килобайтов, гигабайт – это тысяча мегабайтов и т. д. Причем, такие десятичные характеристики сейчас используют и фирмы, поставляющие на рынок системы компьютерной памяти, что приводит к некоторому рассогласованию объявленных в рекламе характеристик с характеристиками, которые мы видим в наших компьютерах.

В нашей статье мы будем пользоваться двоичной шкалой измерения объема информации.

**Уточнение задачи.** Давайте считать, что компьютер Alice к моменту проверки хранит двоичную последовательность

$$x = x_1 x_2 \ldots x_{n-1} x_n$$

из $n$ битов, а компьютер Bob – последовательность

$$y = y_1 y_2 \ldots y_{n-1} y_n$$

из $n$ битов. Наша задача состоит в том, чтобы проверить, выполняется ли $x = y$?

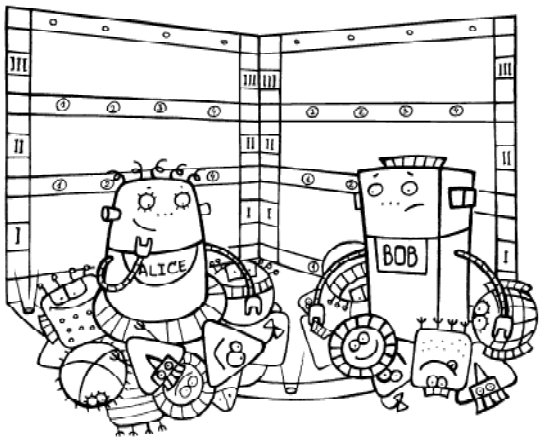
В случае, если сравниваемые на идентичность базы данных имеют размер от терабайта до ста двадцати восьми петабайт, то для последовательностей битов $x$ и $y$ их длина $n$ может быть величиной от $2^3 \cdot 2^{40} = 2^{43}$ до $128 \cdot (2^3 \cdot 2^{50}) = 2^7 \cdot (2^3 \cdot 2^{50}) = 2^{60}$.

*Математиками доказано, что при конструировании детерминированных протоколов по сравнению содержимого компьютеров Alice и Bob требуется передавать n битов информации, и этот объем передачи нельзя уменьшить никакими предварительными вычислениями на компьютерах Alice и Bob.*

*Доказательство этого факта приводится в университетских курсах теории сложности вычислений и подробно излагается в книге* [1].

Вдобавок отметим, что при существующих Интернет-технологиях передача терабайтов и петабайтов информации без ошибок практически нереальна.

Сейчас мы переходим к описанию вероятностных протоколов по сравнению содер-




*...при конструировании детерминированных протоколов по сравнению содержимого компьютеров Alice и Bob требуется передавать n битов информации и этот объем передачи нельзя уменьшить никакими предварительными вычислениями...*

жимого компьютеров Alice и Bob, которые оказываются во много раз эффективнее детерминированных.

## ВЕРОЯТНОСТНЫЙ КОММУНИКАЦИОННЫЙ ПРОТОКОЛ ONE-BIT-WITNESS

Компьютеры Alice и Bob содержат двоичные последовательности $x = x_1x_2...x_{n-1}x_n$ и $y = y_1y_2...y_{n-1}y_n$.

• *Вероятностный этап.* Компьютер Alice равновероятно (с вероятностью $1/n$) выбирает число $i$ из множества $\{1, 2, ..., n\}$.

• *Детерминированный этап.* Компьютер Alice отправляет компьютеру Bob выбранный номер $i$ и один бит (0 или 1), являющийся значением $x_i$ $i$-ого бита своей последовательности $x$.

Компьютер Bob, получив от Alice бит (0 или 1), являющийся значением $x_i$ и номер $i$, выполняет следующие операции:

1. Сравнивает значение полученного бита ($x_i$-го бита) со значением $y_i$ $i$-го бита последовательности $y$.

2. Если $x_i = y_i$, тогда Bob выдает ответ «**последовательности x и y равны**». Если $x_i \neq y_i$, тогда Bob выдает ответ «**последовательности x и y не равны**».

## АНАЛИЗ РАБОТЫ ПРОТОКОЛА ONE-BIT-WITNESS

Рассмотрим два случая. Первый – последовательности $x$ и $y$ одинаковы ($x = y$) и второй — последовательности $x$ и $y$ различны ($x \neq y$).

*Первый случай.* Если последовательности $x$ и $y$ одинаковы, то протокол one-bit-WITNESS всегда выдаст правильный ответ. Действительно, если Alice выбирает число $i \in \{1, ..., n\}$ и отправляет бит $x_i$ компьютеру Bob, то Bob, сравнив биты $x_i$ и $y_i$, выдаст ответ «последовательности $x$ и $y$ равны». Этот ответ будет получен во всех возможных случаях, какой бы номер $i \in \{1, ..., n\}$ не был бы выбран при вероятностном выборе.

Следовательно в первом случае вероятность результата «**последовательности x и y равны**» – единица, а вероятность результата «**последовательности x и y не равны**» – ноль.

*Второй случай.* Подсчитаем вероятность правильной работы протокола one-bit-WITNESS, если $x \neq y$. Рассмотрим два крайних случая:

а) последовательности $x$ и $y$ совпадают во всех битах кроме одного и

б) у последовательностей $x$ и $y$ все биты различны кроме одного.

*Случай а).* Вероятность правильного ответа «**последовательности x и y не равны**» протокола one-bit-WITNESS равна $1/n$, а вероятность ошибки (выдачи ответа «**последовательности x и y равны**») составляет $1 - 1/n$.

*Случай б).* Вероятность правильного ответа «**последовательности x и y не равны**» протокола one-bit-WITNESS равна $\frac{n-1}{n} = 1 - \frac{1}{n}$, а вероятность ошибки составляет $1 - \frac{n-1}{n} = \frac{1}{n}$.

## ОБЪЕМ ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ ПРОТОКОЛОМ ONE-BIT-WITNESS

Для передачи информации от Alice к Bob протоколу one-bit-WITNESS достаточно использовать

$$\lceil \log_2 n \rceil + 1 \qquad (1)$$

битов. Символы $\lceil$ и $\rceil$ в формуле (1) используются для округления до ближайшего большего целого числа. Например, $\lceil 2.3 \rceil = 3$, $\lceil 7.001 \rceil = 8$ и $\lceil 9 \rceil = 9$.

Обсудим формулу (1). Для передачи информации от Alice к Bob протоколу one-bit-WITNESS нужен один бит — значение $x_i$ и номер $i$. Число $i$ выбирается равновероятно из множества $\{1, ..., n\}$. Выберем число $l = \lceil \log_2 n \rceil$. Зафиксируем множество $B^l$ – множество всех двоичных последовательностей длины $l$. Каждому номеру $i \in \{1, ..., n\}$ поставим во взаимно однозначное соответствие двоичную последовательность $code(i)$ из $B^l$. Будем говорить, что последовательность $code(i)$ является кодом номера $i$. Такое кодирование можно выбрать любым удобным нам способом. Так как $|B^l| = 2^l \geq n$, то последовательностей из $B^l$ нам хватит для кодирования. Итак, в соответствии с протоколом one-bit-WITNESS, компьютер Alice выбирает число $i$, определяет код $code(i)$ и отправляет компьютеру Bob один бит (значение $x_i$) и $code(i)$. При этом общая длина сообщения равна $l + 1 = \lceil \log_2 n \rceil + 1$. Наша оценка (1) доказана.

Итак, протокол one-bit-WITNESS – весьма экономный (использует всего $\lceil \log_2 n \rceil + 1$ битов для передачи информации). На одинаковых наборах $x$ и $y$ протокол всегда работает верно. На наборах $x$ и $y$, которые значительно отличаются друг от друга вероятности правильной работы будут удовлетворительными. А вот на наборах $x$ и $y$, которые мало отличаются, протокол one-bit-WITNESS будет ошибаться с большой вероятностью

**Упражнение 1.** Проанализируйте работу протокола one-bit-WITNESS, если $n = 10^{16}$, а последовательности $x$ и $y$ различны в $10^5$ битах.

Подсчитайте вероятности правильной работы протокола one-bit-WITNESS, если 9/10 или 3/4 долей последовательностей $x$ и $y$ различны.

Проверьте вероятности правильной работы, если 9/10 или 3/4 долей последовательностей $x$ и $y$ одинаковы.

## ВЕРОЯТНОСТНЫЙ КОММУНИКАЦИОННЫЙ ПРОТОКОЛ ONE-BIT-WITNESS(10)

Определим вероятностный протокол one-bit-WITNESS(10) – модификацию протокола one-bit-WITNESS. Работа one-bit-WITNESS(10) состоит в том, что он несколько раз проводит серию независимых проверок $x = y$, запуская коммуникационный протокол one-bit-WITNESS.

Если все варианты проверок заканчиваются результатом «**последовательности *x* и *y* равны**», то ответ протокола one-bit-WITNESS(10) будет «**последовательности *x* и *y* равны**».

Если хотя бы один вариант проверки заканчивается результатом «**последовательности *x* и *y* не равны**», то ответ протокола one-bit-WITNESS(10) будет «**последовательности *x* и *y* не равны**».

**Упражнение 2.** Подсчитайте число передаваемых бит и вероятность правильной работы протокола one-bit-WITNESS(10), повторяющего сто раз работу протокола one-bit-WITNESS, если $n = 10^{16}$, а последовательности $x$ и $y$ различны в $10^5$ битах.

Сравните, как изменятся вероятности результатов работы протокола one-bit-WITNESS(10) по сравнению с протоколом one-bit-WITNESS на этих последовательностях.

## АНАЛИЗ РАБОТЫ ПРОТОКОЛА ONE-BIT-WITNESS(10)

Поскольку протокол one-bit-WITNESS(10) запускает работу протокола one-bit-WITNESS 10 раз, общее число пересылаемых бит равно $10 \cdot (\lceil \log_2 n \rceil + 1)$.

Общий итог рассмотрения протокола one-bit-WITNESS и его модификации one-bit-WITNESS(10) не утешителен. При требованиях высокой надежности и экономности передачи информации они не подходят. Повторения протокола one-bit-WITNESS увеличивают вероятности правильного результата, однако если число повторений $k$ слишком велико (например близко к $n$), то мы сильно ухудшаем время работы и силь-

но увеличиваем общий объем пересылаемой информации.

Можно ли разработать вероятностный протокол, который по числу пересылаемой информации сравним с протоколом one-bit-WITNESS и при этом во всех случаях гарантирует большую надежность? Ответ, да, такой протокол разработан. Его мы сейчас опишем.

Предварительно введем следующие соглашения и обозначения. Будем интерпретировать последовательности $x = x_1...x_n$ и $y = y_1...y_n$ как двоичные представления целых чисел

$$Number(x) = \sum_{i=1}^{n} 2^{n-1} \cdot x_i ,$$

$$Number(y) = \sum_{i=1}^{n} 2^{n-1} \cdot y_i .$$

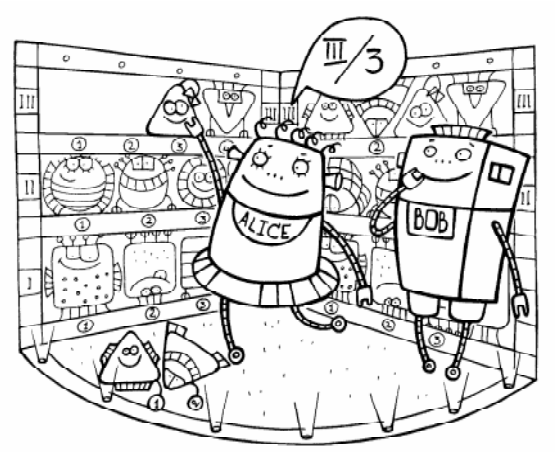
Обозначим через $PRIM(m)$ множество всех простых чисел, не превышающих число $m$:

$$PRIM(m) = \{p \text{ простое число}: \ p \le m \}$$

Напомним, что целое число $p$ называется простым, если оно не делится нацело ни на какие числа кроме него самого и единицы. Через $|PRIM(m)|$ обозначим число элементов в множестве $PRIM(m)$.

## ВЕРОЯТНОСТНЫЙ КОММУНИКАЦИОННЫЙ ПРОТОКОЛ WITNESS

Компьютеры Alice и Bob содержат двоичные последовательности $x = x_1...x_{n-1}x_n$ и $y = y_1...y_{n-1}y_n$. Alice образует для себя вспомогательное множество $PRIM(n^2)$.




*Компьютер Alice отправляет компьютеру Bob выбранное число p и вычисленное число s.*

• *Вероятностный этап.* Компьютер Alice равновероятностно (с вероятностью $\frac{1}{|PRIM(n^2)|}$) выбирает число $p$ из множества $PRIM(n^2)$.

• *Детерминированный этап.* Alice вычисляет число $s$ – остаток от деления числа $Number(x)$ на число $p$. Компьютер Alice отправляет компьютеру Bob выбранное число $p$ и вычисленное число $s$.

Компьютер Bob, получив числа $p$ и $s$, производит следующие действия:

1. Вычисляет остаток $q$ от деления числа $Number(y)$ на число $p$.

2. Если $s = q$, тогда Bob выдает ответ **«последовательности x и y равны»**. Если $r \neq s$, тогда Bob выдает ответ **«последовательности x и y не равны»**.

Как видите, протокол WITNESS достаточно прост в своем описании. Мы представим анализ объема передаваемой информации и подсчет вероятности правильной работы протокола в общем случае в следующей статье. Сейчас мы проиллюстрируем работу WITNESS на конкретном примере, который позволит нам полнее прочувствовать конструкцию и математическое изящество описанного протокола WITNESS.

**Пример 1.** Пусть $n = 6$. Пусть Alice и Bob содержат последовательности $x = 001111$ и $y = 010110$. Итак имеем:

$$Number(x) = 2^3 + 2^2 + 2^1 + 2^0 = 15,$$

$$Number(y) = 2^4 + 2^2 + 2^1 = 22,$$

$$PRIM(6^2) = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31\}.$$

На первом этапе работы Alice равновероятностно (с вероятностью 1/11) выбирает один из 11 возможных (детерминированных) путей вычисления, соответствующих простым числам из множества $PRIM(6^2)$.

Предположим, что Alice выбирает 5 ($p = 5$). Alice вычисляет остаток $s$ от деления 15 на 5 ($s = 0$) и отправляет Bob числа $p$ и $s$.

Bob, получив числа $p$ и $s$, вычисляет остаток $q$ от деления 22 на 5 ($q = 2$). Далее

Bob сравнивает числа $s$ и $q$. Так как $2 = q \neq s = 0$, то Bob выдает правильный ответ

**«последовательности x и y не равны»**

Теперь предположим, что Alice выбирает число 7 из $PRIM(6^2)$ ($p = 7$). Тогда, в соответствии с протоколом, компьютер Alice вычисляет остаток $s$ от деления 15 на 7 $s = 1$ и отправляет Bob числа $p$ и $s$. Получив числа $p$ и $s$, Bob вычисляет остаток $q$ от деления 22 на ($q = 1$). Так как в данном случае оказывается, что $s = q = 1$, то Bob выдает неправильный ответ

**«последовательности x и y равны»**.

Итак, в нашем примере имеются варианты правильной и неправильной обработки последовательностей $x$ и $y$ протоколом WITNESS.

Давайте проанализируем наш пример.

**Упражнение 3.** Пусть $x$ = 001111 и $y$ = 010110.

1. Существует ли в множестве $PRIM(6^2)$ простое число, отличное от 7, выбор которого привел бы к неправильному ответу протокола WITNESS?

2. Сколько таких «плохих» простых чисел в $PRIM(6^2)$ для наших двух последовательностей $x, y$?

3. Какова вероятность правильной обработки последовательностей $x, y$ протоколом WITNESS?

4. Подсчитайте, сколько бит достаточно передать от компьютера Alice к компьютеру Bob при работе протокола WITNESS.

**Упражнение 4.**

Пусть теперь $x = y$ = 100110.

Существует ли простое число из $PRIM(6^2)$, выбор которого приведет протокол WITNESS к неправильному результату **«последовательности x и y не равны»**?

**Упражнение 5.**

Пусть $x$ = 10011011 и $y$ = 01010101.

1. Подсчитайте, сколько простых чисел из $PRIM(8^2)$ приведут протокол WITNESS к неправильному ответу **«последовательности x и y равны»**, а сколько – к правильному **«последовательности x и y не равны»**.

2. Какова вероятность правильной обработки последовательностей $x, y$ протоколом WITNESS?

3. Решите сколько бит достаточно передать от компьютера Alice к компьютеру Bob при работе протокола WITNESS.

**Упражнение 6.**

Слово «witness» с английского языка переводится, как «свидетель», «очевидец». Как вы считаете почему мы наш протокол называем WITNESS?

В следующей статье мы приведем анализ нашего протокола WITNESS и рассмотрим решения упражнений. Желаем удачи!

**Литература**

1. *Hromkovic J.* Communication complexity and parallel computations, Springer Verlag Press, 1997.

***Юрай Громкович,***
***Professor of Computer Science, Swiss Federal Institute of Technology, Zürich,***

***Аблаев Фарид Мансурович,***
***доктор физико-математических наук, профессор, заведующий кафедрой теоретической киберенетики Казанского федерального университета.***