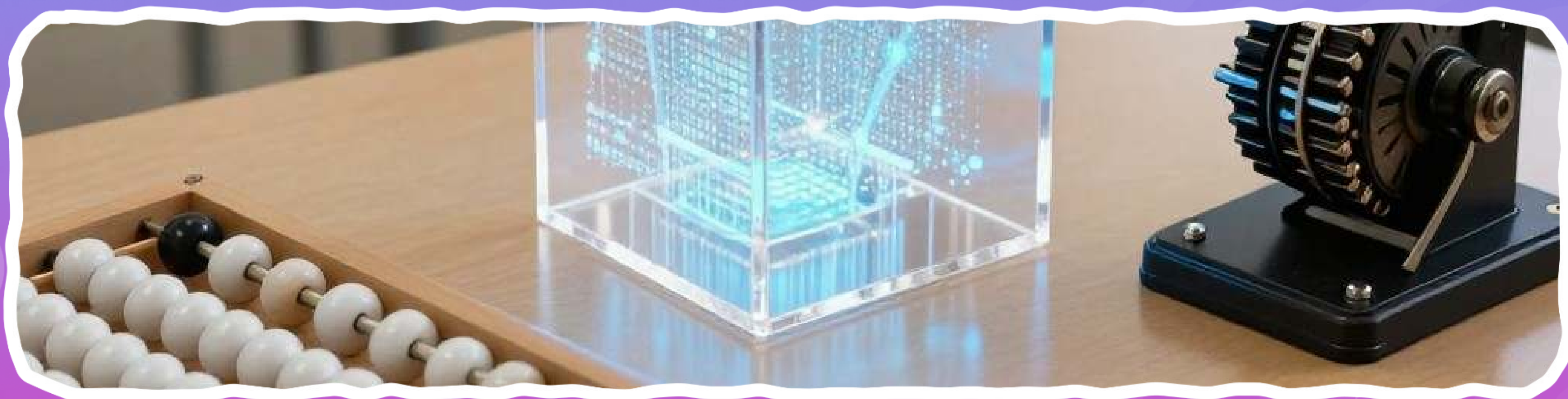


Математические основы информатики

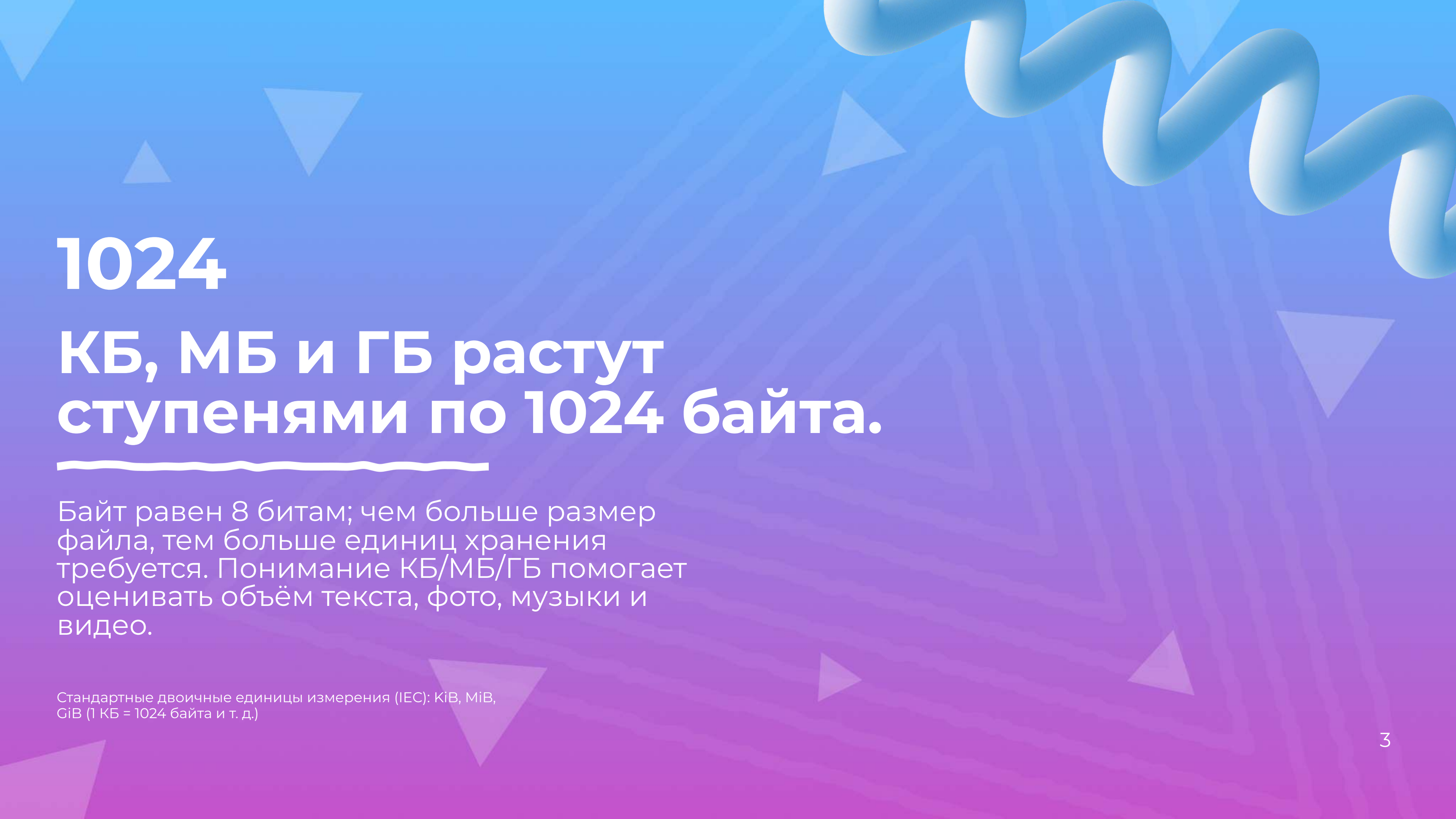
Информатика изучает информацию и её обработку; математика помогает хранить, кодировать и защищать данные.



Зачем восьмикласснику нужны математические идеи в информатике

Компьютер работает с данными как с двоичными числами, поэтому важны системы счисления. Логика нужна для условий и фильтров, графы — для маршрутов и связей, а алгоритмы задают точный порядок действий.





1024

КБ, МБ и ГБ растут степенями по 1024 байта.

Байт равен 8 битам; чем больше размер файла, тем больше единиц хранения требуется. Понимание КБ/МБ/ГБ помогает оценивать объём текста, фото, музыки и видео.

Стандартные двоичные единицы измерения (IEC): KiB, MiB, GiB (1 КБ = 1024 байта и т. д.)

Единицы хранения данных и примеры

В таблице — основные шаги от бита до мегабайта. Остальные единицы: 1 ГБ — фильм или крупная игра, 1 ТБ — архив из множества файлов.

Единица	Сколько байт	Пример
1 бит	—	минимум
1 байт	1	символ
1 КБ	1024	короткий абзац
1 МБ	1024 КБ	фото среднего качества

Чем выше единица, тем быстрее растёт объём данных; поэтому важно понимать точные соотношения при работе с файлами.

Стандартные двоичные единицы измерения информации (KiB, MiB, GiB, TiB).

Двоичная система счисления

Почему в компьютере «только 0 и 1»

Компьютеры используют всего два состояния, которые удобно различать электрически: сигнал есть или сигнала нет. Поэтому числа записываются с помощью разрядов 0 и 1, а электронные схемы работают надёжно.



Как читать двоичную запись

Двоичная запись показывает, какие степени двойки входят в число. Например, $5 = 101_2$, $8 = 1000_2$, $10 = 1010_2$. Каждый разряд связан со своей степенью двойки.

Сравнение десятичной и двоичной записи чисел



Самостоятельный расчёт длины двоичной записи по степени двойки (n).

Когда число достигает степеней двойки, количество двоичных разрядов увеличивается: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$.

Двоичная запись обычно длиннее десятичной, зато хорошо подходит для устройств, где работают два устойчивых состояния.

Перевод чисел между системами счисления

01

Из десятичной системы в двоичную удобнее переводить делением на 2: остатки сохраняют. Когда деление доводит число до нуля, остатки читают снизу вверх.

02

Чтобы перевести из двоичной в десятичную, считают вклад разрядов: суммируют степени двойки, где стоит 1. Так получаются адреса, коды символов и значения параметров устройств.

03

Обе процедуры логичны и проверяемы: первую можно выполнить в столбик, вторую — как «подсчёт включённых степеней». Это помогает понимать, что именно хранит компьютер.

Как число превращается в двоичный код

Преобразование показано на примере десятичного числа (например, 13) делением на 2 и чтением остатков.



Алгебра логики в информатике



Истина и ложь

Логика рассматривает высказывания, которые могут быть истинными или ложными. В информатике для них используют значения 1 и 0, чтобы компьютеру было легче выполнять проверки.



Операции НЕ, И, ИЛИ

Операция НЕ меняет значение на противоположное. И даёт 1 только при двух единицах, а ИЛИ — при наличии хотя бы одной единицы. Эти правила формируют смысл условий.



Где применяются выражения

Логические выражения встречаются в программах, играх, поиске и проверке условий. Они помогают компьютеру решать, какое действие выполнить при разных вариантах данных.

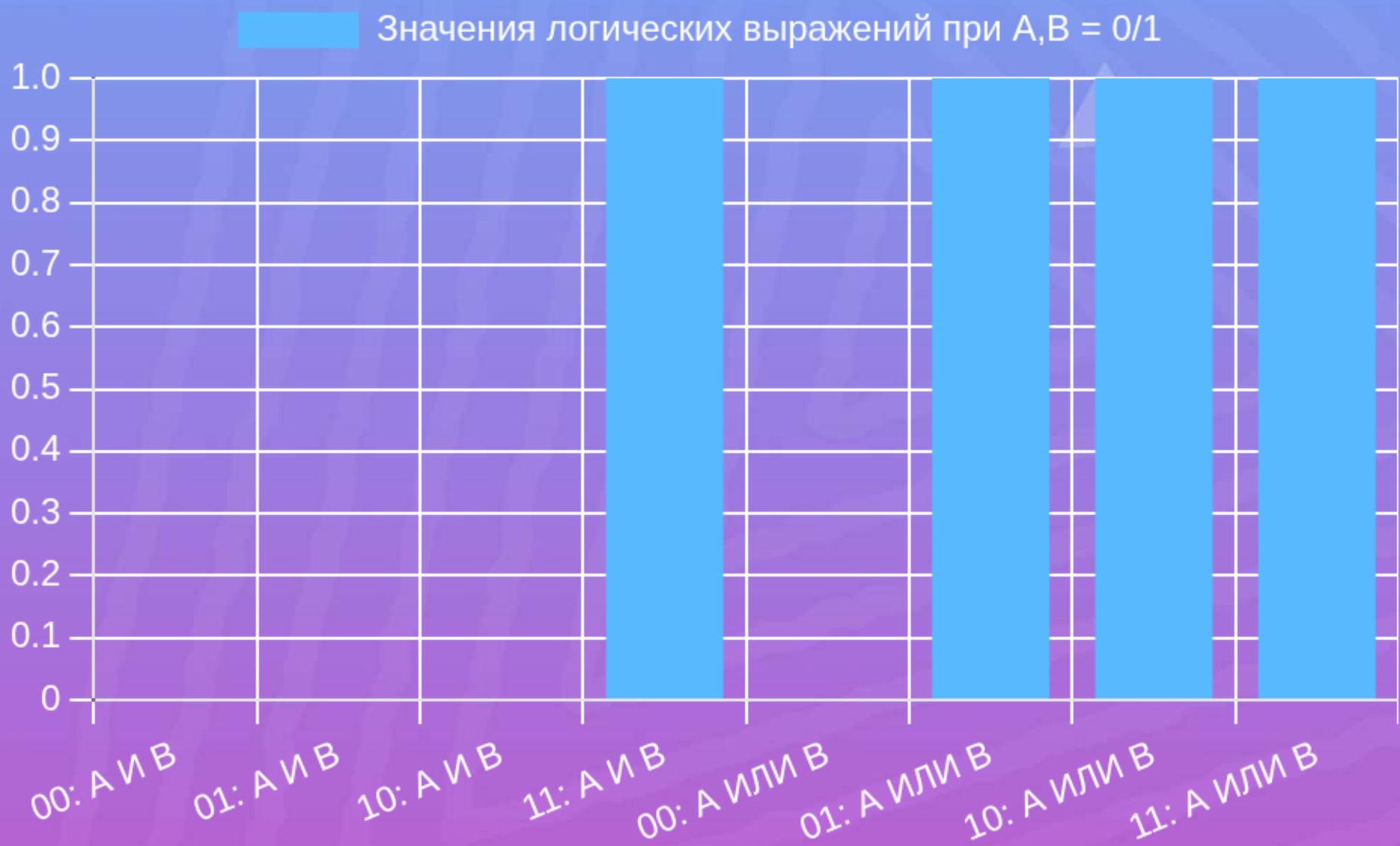
Логические операции и таблицы истинности

А	В	А И В	А ИЛИ В	НЕ А
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Таблица показывает результат выражений для всех комбинаций 0 и 1. Это точный способ проверить, как логика ведёт себя при разных значениях А и В.

По таблице видно различие между строгим условием (И) и более широким выбором (ИЛИ), а НЕ просто переворачивает значение.

Результаты логических выражений для всех комбинаций



Для набора 11 выражение A И B даёт 1, а при остальных комбинациях — 0. Это показывает строгий характер операции «и» и её зависимость от двух истин одновременно.

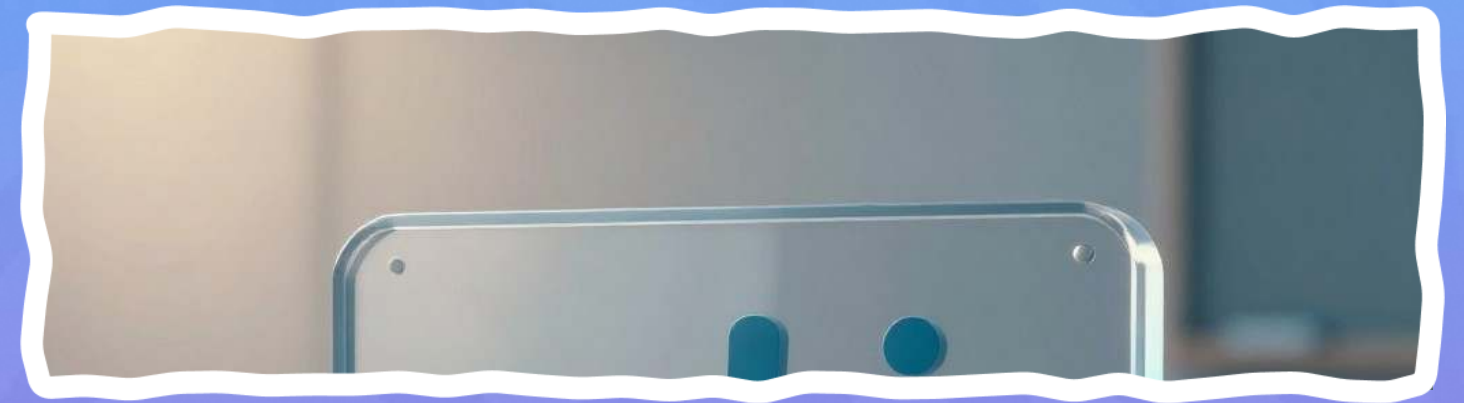
Диаграмма подчёркивает разницу между «строгим условием» (только 11) и «широким выбором» (три случая из четырёх).

Таблица истинности для операций НЕ, И, ИЛИ; расчёт значений для комбинаций 00, 01, 10, 11.

Как код превращает символы и картинки в числа

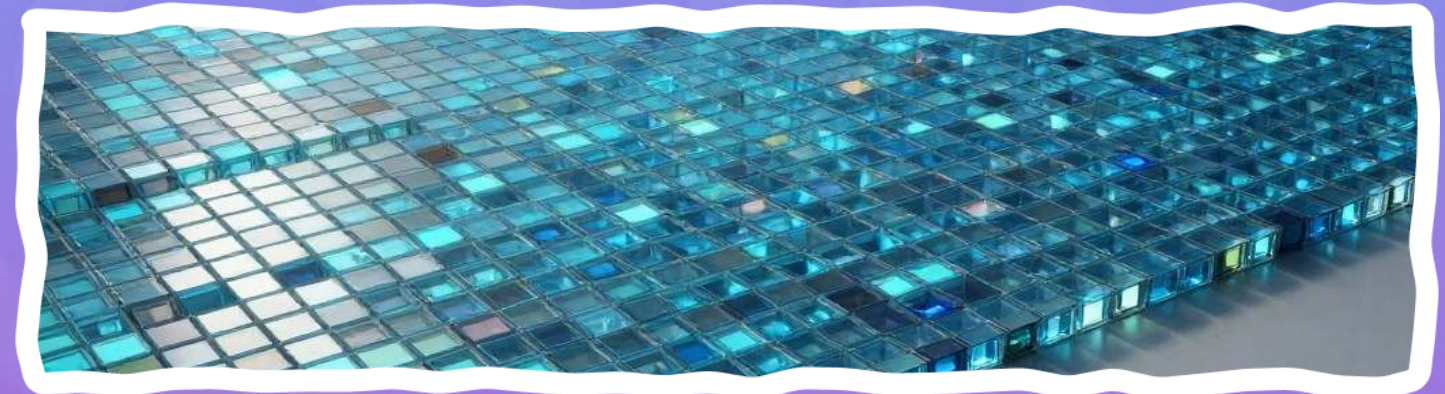
Один символ — один номер

Каждый знак, который ты видишь в сообщении, превращается в числовой код по таблице кодировки. Например, буква, цифра или знак препинания получают свой номер. Компьютер хранит эти номера, а программу показывает символы обратно.



Сетка пикселей как «лист клеток»

Изображение состоит из пикселей, а каждый пиксель хранит цвет как число. Если сделать сетку плотнее, деталей станет больше. Тогда на хранение потребуется больше информации: больше значений цветов для всех точек.

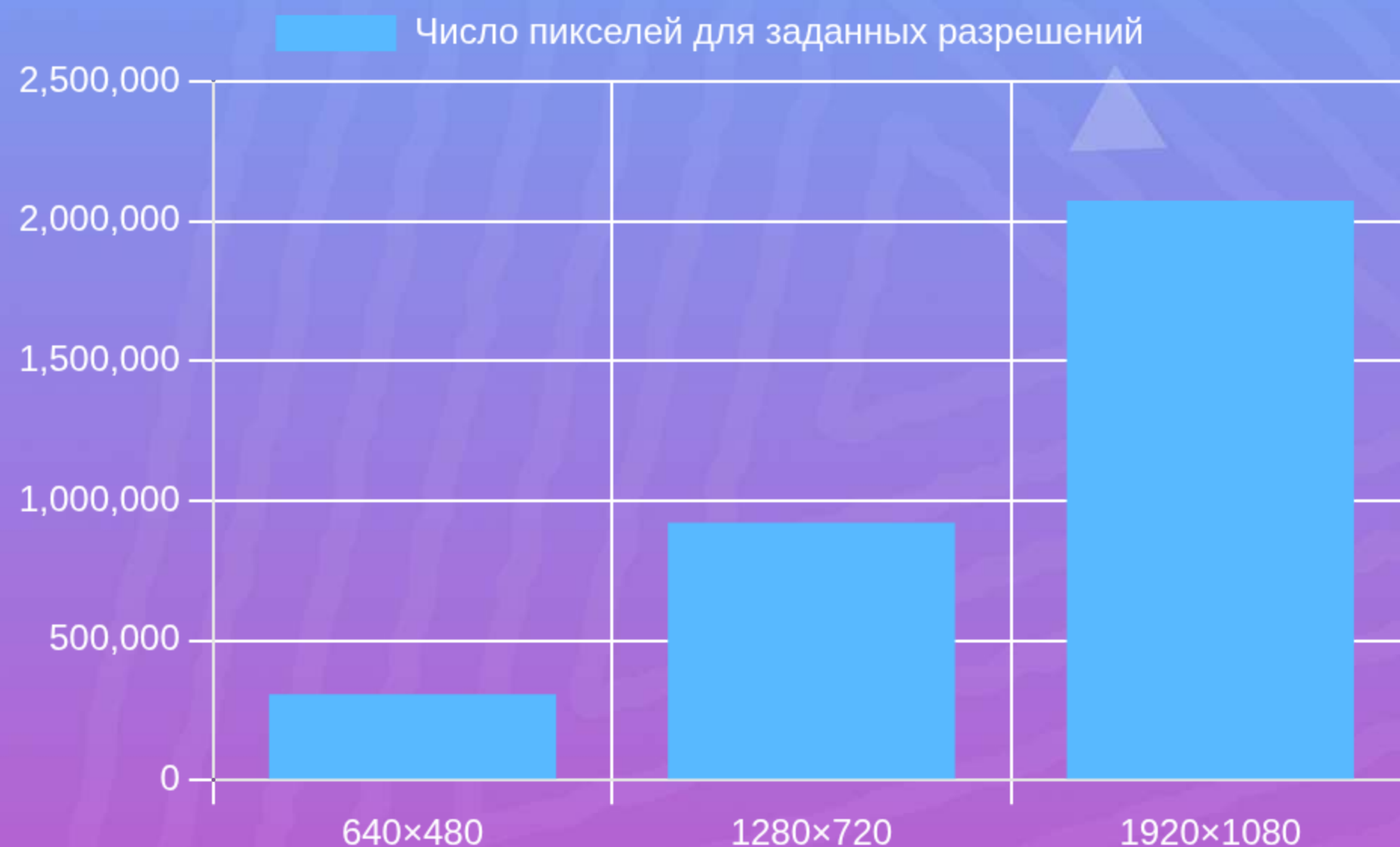


Разрешение и память растут вместе

Когда меняется разрешение, меняется и количество пикселей: больше точек — больше чисел для записи. Кроме того, чем больше оттенков поддерживает цветовая схема, тем больше данных нужно на каждый пиксель. Поэтому файл становится тяжелее.



Как разрешение влияет на объём изображения



Расчёт по формуле ширина×высота; использованы разрешения 640×480, 1280×720, 1920×1080.

У каждого разрешения число пикселей равно произведению ширины на высоту. Поэтому рост «ширины» и «высоты» усиливает эффект почти пропорционально.

По данным диаграммы видно: при увеличении разрешения количество точек растёт очень быстро, и вместе с ним обычно увеличивается размер файла.

Алгоритм как точная последовательность действий

01

Алгоритм — это конечная последовательность шагов, где каждому действию соответствует понятный результат. Если записать шаги ясно и без «как-нибудь», компьютер сможет выполнить задачу без ошибок.

02

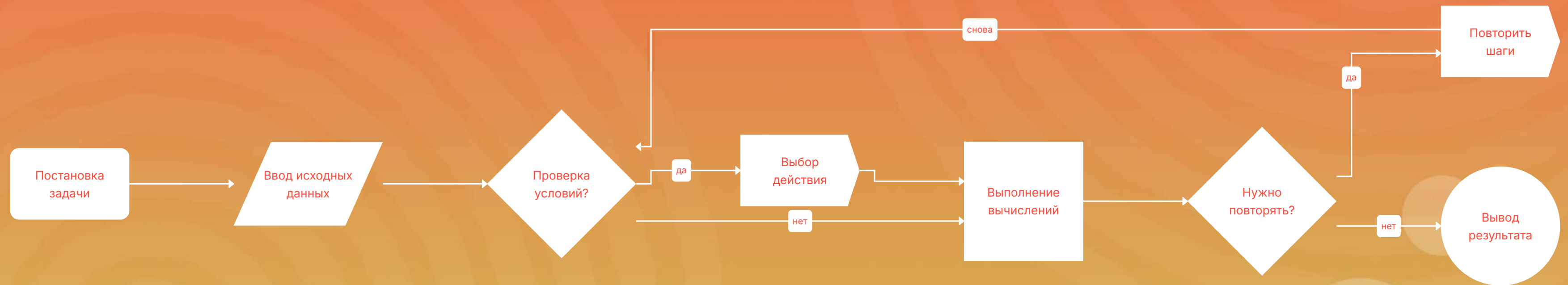
Свойства алгоритма помогают проверять качество записи: определённость снимает двусмысленность, результативность гарантирует получение ответа, массовость позволяет применять решение к разным входным данным.

03

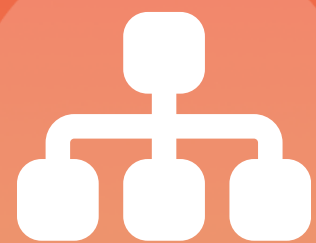
Конечность означает, что процесс останавливается. В информатике алгоритмы описывают вычисления и проверки, а математика помогает убедиться, что переходы не противоречат друг другу.

Пошаговое решение задачи: от условий к результату

Типовая схема алгоритма: ветвления для условий и циклы для повторений.



Графы и сети: как связи становятся моделью



Вершины — объекты

Вершина в графе обозначает отдельный объект: город, пользователя, сайт или точку маршрута. Она нужна, чтобы у каждой «сущности» было своё место в модели.



Рёбра — связи

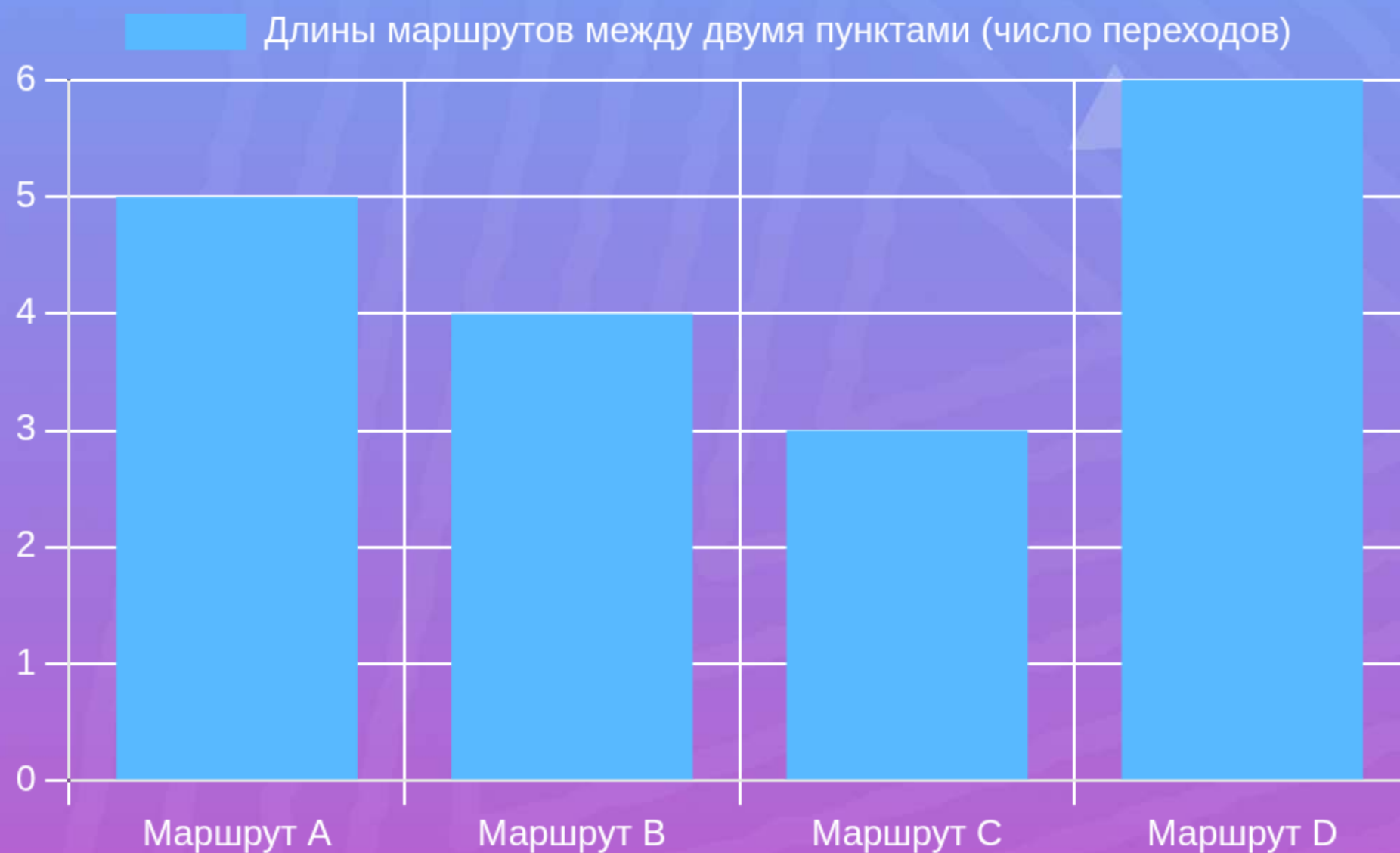
Рёбра показывают, какие объекты связаны между собой. Например, дорога между городами или канал связи в интернете. Направленные рёбра помогают различать «куда» и «откуда».



Путь и выбор маршрута

По графу можно искать путь: последовательно проходить по рёбрам от одной вершины к другой. Это помогает проверять наличие связи и выбирать маршрут с нужными свойствами.

Поиск кратчайшего пути на графе



Примерная схема сравнения маршрутов по длинам ребер;
конкретные значения зависят от заданного графа.

На рисунке пути различаются числом переходов или общей длиной рёбер. Кратчайший вариант отмечают отдельно, чтобы сравнение было наглядным.

Минимальное значение соответствует самому короткому пути. Такой выбор уменьшает число шагов, а значит обычно экономит время и ресурсы.

Сложность алгоритмов

01

Сложность показывает, как меняется количество шагов при росте объёма данных. Если входных значений становится больше, алгоритм может выполнять всё больше операций.

02

Рост не всегда «линейный»: при увеличении данных в 10 раз время работы способно вырасти в 100 раз и более. Причина — как именно алгоритм перебирает и сравнивает варианты.

03

Простой перебор часто делает много проверок, поэтому его работа быстро замедляется на больших наборах. Более продуманные методы могут находить ответ быстрее, выполняя меньше действий.

04

Оценивая сложность, можно выбирать подходящий алгоритм для школьных задач и реальных сервисов. Для больших данных важна не только правильность, но и скорость решения.

Математика в повседневной информатике

Поиск в интернете опирается на логические условия и слова запроса: фильтры «включить/исключить» сужают результаты. Навигация на карте использует графы и идеи кратчайшего пути, чтобы предложить маршрут.

Сжатие файлов использует кодирование, уменьшая количество бит, которые нужно хранить или передавать. В играх важны координаты, счёт и модели вероятности, а надёжность паролей растёт вместе с числом возможных комбинаций.

Математика делает информатику точной и понятной

Системы счисления, логика, кодирование, графы и алгоритмы формируют основу информатики. Благодаря этим идеям компьютер хранит данные, выполняет команды, ищет информацию и решает задачи — уверенно и предсказуемо.

