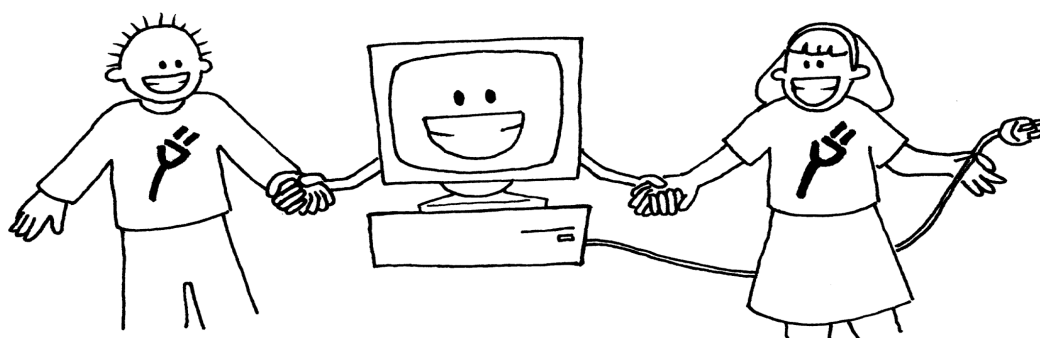


# COMPUTER SCIENCE *Unplugged*

Тим Белл, Ян Х. Виттена, Майк Феллоус

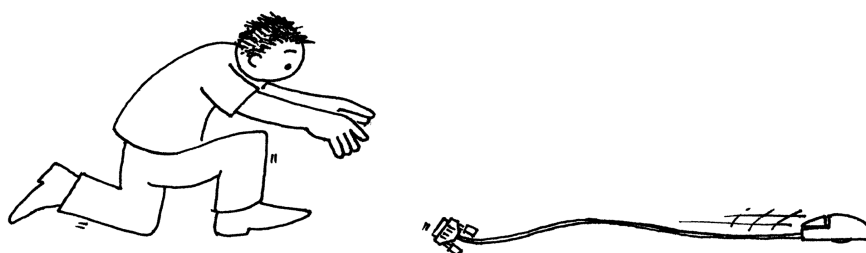


Методические рекомендации, для использования в классе, подготовили

Робин Адамс и Джейн Макензи

Рисунки выполнены Мэттом Пауэллом

Перевод с английского Ирины А. Деревянко





# Предисловие

---

Компьютеры мы встречаем везде. Поэтому нам необходимо научиться использовать их, а многие из вас уже используют их каждый день. Но как работает компьютер? Как он думает? И как людям сделать так, чтобы компьютеры работали быстрее и эффективнее? Информатика увлекательный предмет, который поможет вам ответить на эти вопросы. Легко и интересно, в процессе деятельности, эта книга познакомит вас с работой компьютера без самого компьютера.

Эта книга может быть использована вами как на обычных учебных занятиях, так и во внеурочное время. Вам не обязательно быть крупным специалистом по компьютерам, чтобы применить предложенные нами принципы в обучении ваших детей. Книга состоит из определенных видов деятельности, основанных на принципах работы компьютера. Справочная информация позволит вам легко освоить каждый вид деятельности. Все задания имеют ответы, и каждая деятельность завершается вопросом: «О чем это всё?» Это раздел, в котором объясняются важные сведения изучаемого вами вида деятельности.

Некоторые виды деятельности можно назвать математическими основами информатики: функция и графы, математические модели и задачи сортировки, а также криптография. Другим направлением деятельности является информационно-коммуникационные технологии, рассматривающие проблемы работы самого компьютера и различных учебных программ. Дети активно общаются, решают различные творческие задачи, развивают навыки мышления в данной области знания.

В дополнение к этой книге, данный проект «Unplugged», имеет большое количество свободно распространяемого электронного ресурса: видео, рисунки и различный дополнительный материал на «[csunplugged.org](http://csunplugged.org)».

Эта книга была написана тремя преподавателями информационных технологий и двумя учителями, и основана на опыте работы с детьми в различных классах. Мы находим, что многие важные понятия информатики можно и нужно вводить без использования компьютера, т.к. иногда компьютер отвлекает от обучения.

А сейчас отключите свой компьютер, и готовьтесь узнать, все об информатике и компьютерах.

Эта книга доступна для свободного использования в личных и образовательных целях благодаря грантам от Google, Inc. Она распространяется по «Creative Commons Attribution-Noncommercial-NoDerivs» лицензии, это означает, что вы можете свободно копировать, распространять и отображать Книгу при условии, что вы не вносите никаких изменений в содержание (в том числе, нельзя присваивать авторство и изменять условия лицензии), и вы не имеете право использовать эту книгу в коммерческих целях. Мы рекомендуем использование этого материала в учебных заведениях, а также, вам разрешается распечатать свой экземпляр книги и распространять отдельные листы среди своих учеников. Мы рады будем получить от вас предложения по переводу на другие языки, которые вы можете направить авторам (см. [csunplugged.org](http://csunplugged.org)).

# Благодарности

---

Много детей и учителей помогли нам скорректировать наши планы. Дети и учителя в South Park School (Виктория, Британская Колумбия), Shirley Primary School, Ilam Primary School and Westburn Primary School (Christchurch, Новая Зеландия) были подопытными кроликами для многих видов деятельности. Мы особенно благодарны Linda Picciotto, Karen Able, Bryon Porteous, Paul Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield, and Lynn Atkinson, которые радушно принимали нас в своих классах и давали полезные советы для улучшения учебной деятельности. Гуэнда Bensemman апробировала некоторые виды учебной деятельности, и внесла важные изменения. Ричард Lynders и Sumant Murugesh помогли с экспериментальным классом. Некоторые разделы криптографической учебной деятельности были разработаны Кеном Noblitz. Часть из этой деятельности проводилась под эгидой Victoria "Mathmania" групп, с помощью Kathy Beveridge. Более ранние версии рисунков были сделаны Malcolm Robinson и Gail Williams, и мы также воспользовались советом Hans Knutson.

Matt Powell также оказывал неоценимую помощь в развитии "Unplugged" проекта. Мы благодарны Brian Mason за научную и техническую поддержку и щедрую спонсорскую помощь на ранних стадиях написания этой книги. Особую благодарность Paul и Ruth Ellen Howard, которые испытали много видов данной деятельности и высказали ряд полезных замечаний.

Peter Henderson, Bruce McKenzie, Joan Mitchell, Nancy Walker-Mitchell, Gwen Stark, Tony Smith, Tim A. H. Bell<sup>1</sup>, Mike Hallett, and Harold Thimbleby также представили много полезных замечаний.

Мы благодарны нашим родным: Bruce, Fran, Grant, Judith, и Pam за их поддержку, и Andrew, Anna, Hannah, Max, Michael, and Nikki, которые являлись вдохновителями большей части этой работы<sup>2</sup>, и часто были первыми тестировщиками этой деятельности.

Мы особенно благодарны компании Google Inc за финансирование проекта «Unplugged», что позволяет нам сделать это издание доступным для свободного скачивания. Мы приветствуем замечания и предложения касательно каждой учебной деятельности. С авторами можно связаться через «csunplugged.org».

---

<sup>1</sup> Однофамилец автора книги.

<sup>2</sup> В самом деле, учебная деятельность: сжатие текста, была придумана Michael.

# Содержание

---

Введение	Ошибка! Закладка не определена.
Благодарности	ii
<b>Данные: строительный материал - Представление информации</b>	<b>1</b>
Подсчёт точек — <i>Двоичные числа</i>	3
Цвет Чисел — <i>Представление изображения</i>	14
Вы сможете это повторить! — <i>Сжатие текста</i>	23
Фокус с карточками — <i>Обнаружение ошибки &amp; Исправление</i>	31
Двадцать предположений — <i>Теория информации</i>	37
<b>Заставим Компьютеры работать — Алгоритмы</b>	<b>43</b>
Кораблики — <i>Алгоритмы поиска</i>	45
Самый легкий или самый тяжёлый — <i>Алгоритмы сортировки</i>	64
Бой часов — <i>Сортировка сетей</i>	71
Грязный город — <i>Минимальный обход деревьев</i>	76
Апельсиновая игра — <i>Маширутизация и Тупик в Сетях</i>	81
<b>Укажем Компьютерам что делать — Представление Процедур</b>	<b>84</b>
Охота за сокровищами — <i>Конечный Автомат</i>	86
Команда вперед — <i>Языки программирования</i>	101



# Часть I

**Данные: строительный материал**

*Представление информации*

# Данные: строительный материал

---

## Как можно хранить информацию в компьютерах?

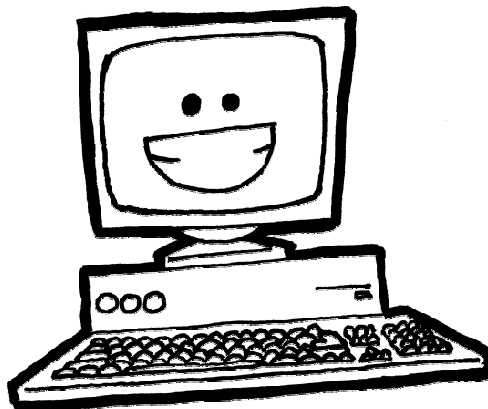
Слово компьютер происходит от латинского «computare», что означает вычислить или сложить или, одновременно и вычислить и сложить, но компьютер сегодня больше, чем просто большой калькулятор. Он может быть: библиотекой, помощником как в письме и в поиске информации, так и в прослушивании музыки и, даже, в просмотре фильмов. Так как же компьютеры хранят всю эту информацию? Верьте или нет, компьютер оперирует только двумя числами: нулем и единицей!

## Чем отличаются данные от информации?

Данные – это строительный материал - числа, с которыми работает компьютер. Компьютер преобразует данные в информацию (слова, числа и изображения), что бы вы эту информацию могли понять.

## Как числа, буквы, слова и картинки получают из нулей и единиц?

В этом разделе вы узнаете о двоичных числах; как компьютер может рисовать; как работает факс; какой существует эффективный способ хранения большого объема информации; что необходимо сделать, чтобы не допустить ошибок; как измерить количество информации, чтобы можно было ее хранить.





# Деятельность 1

---

## Подсчёт точек— Двоичные числа

### Введение

Данные в компьютере хранятся и передаются как ряд нулей и единиц. Как можно представить слова и числа, используя только эти две цифры?

### Учебные направления

- ✓ По математике: От 2 уровня «Числа». Исследование чисел в десятичной и двоичной системе. Представление десятичных чисел двоичными числами.
- ✓ По математике: От 2 уровня «Алгебра». Уметь продолжить действие по предложенному образцу и описать правила этого действия. Отношения и последовательность представления степени с основанием два.

### Навыки

- ✓ Вычисления
- ✓ Сравнения
- ✓ Установление последовательности

### Возраст

- ✓ От 7 лет и старше

### Материалы

- ✓ Вы должны приготовить для показа пять «двоичных» карточек (см. стр. 6). На бумагу формата А4 наклейте крупные точки.

Каждому ребенку необходимо:

- ✓ Набор из пяти карточек. Вырежете карточки из «Фотокопии для учителя: Двоичные числа» (стр. 6).
- ✓ Лист деятельности: «Двоичные числа» (стр. 5)

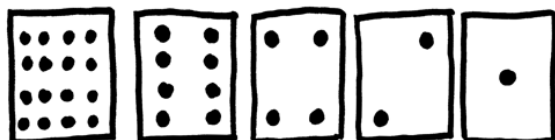
Расширенной деятельности в этом разделе, каждому ребенку необходимо:

- ✓ Лист деятельности: Работа с двоичными числами (стр. 7)
- ✓ Лист деятельности: Отправление секретных сообщений (стр. 8)
- ✓ Лист деятельности: Электронная почта и Модемы (стр. 9)
- ✓ Лист деятельности: Подсчёт чисел больше, чем 31 (стр. 10)
- ✓ Лист деятельности: Подробнее о двоичных числах (стр. 11)

# Двоичные числа

## Инструкция для учителя

Перед тем как раздать листы, на стр. 5, необходимо ознакомиться с принципами работы с группой детей. Для этого вида деятельности необходим набор из пяти карточек, как показано ниже, с точками только на одной стороне. Для демонстрации карточек пригласите пятерых детей из группы, каждому дайте по карточке. И расположите карточки в следующем порядке:



## План беседы

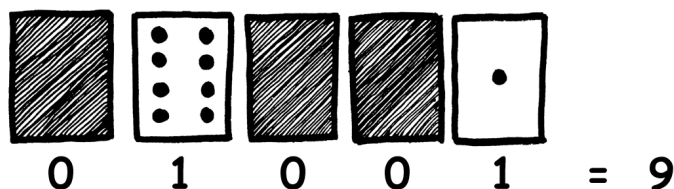
Что вы можете сказать о количестве точек на карточках? (На каждой последующей карточке вдвое больше точек, чем на предыдущей, расположенной правее).

Сколько точек было бы на следующей карте слева, если бы мы продолжили ряд карточек? (32) Поясните подробнее...?

Мы можем использовать эти карточки для показа чисел, переворачивая некоторые карточки на другую сторону и подсчитывая оставшиеся точки. Попросите детей показать число 6 (карточки с 4 точками и 2 точками), потом число 15 (карточки с 8-, 4-, 2- и 1- точками), потом число 21 (16, 4 и 1)... А сейчас попробуйте посчитать с нуля и далее.

Остальные дети в классе должны внимательно наблюдать за переворачиванием карточек и увидеть закономерность (каждая карточка может находиться в двух положениях или с изображенными точками или перевернутая). Вы можете привести другие примеры с показом чисел и привлечь большее количество детей из группы.

Когда карточка перевернута так, что изображение точек отсутствует, то это событие представим как ноль. Событие, когда на карточке изображены точки – представим как один. Таким образом, нами введена двоичная система счисления.



Спросите детей, как показать 01001. Какое это число в десятичной системе? (9) Как будет представлено число 17 в двоичной системе? (10001). Приведите еще несколько примеров. Попробуйте расширить деятельность, для закрепления навыка, до тех пор, пока каждый ребенок из группы не поймет.

## Лист деятельности: Двоичные числа

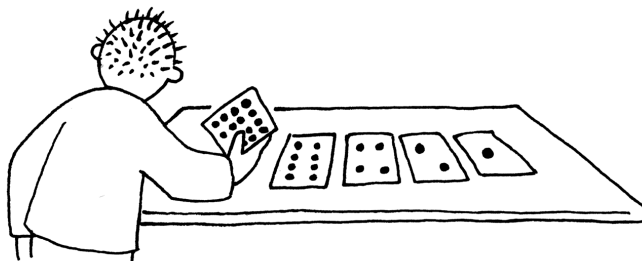
### Учимся счёту

Итак, вы думаете, что уже узнаете как считать? Хорошо, а вот другой способ как это сделать!

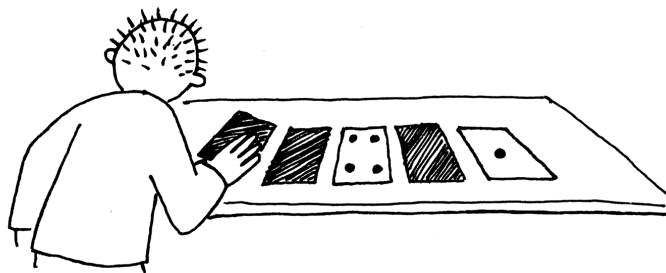
Знаете ли вы, что компьютер использует только ноль и единицу? Все, что вы можете увидеть или услышать на компьютере: слова, рисунки, числа, фильмы и даже звук, хранится в нем с использованием только этих двух чисел! Мы будем сейчас учить вас, как можно отправить секретное сообщение друзьям, используя тот же метод, что и компьютер.

### Указания

Вырежете карточки на вашем листе и положите карточку с 16-тью точками слева, как показано здесь:



Убедитесь, что карточки расположены также как на рисунке. Сейчас переверните карточки так, чтобы получить изображение 5 точек.

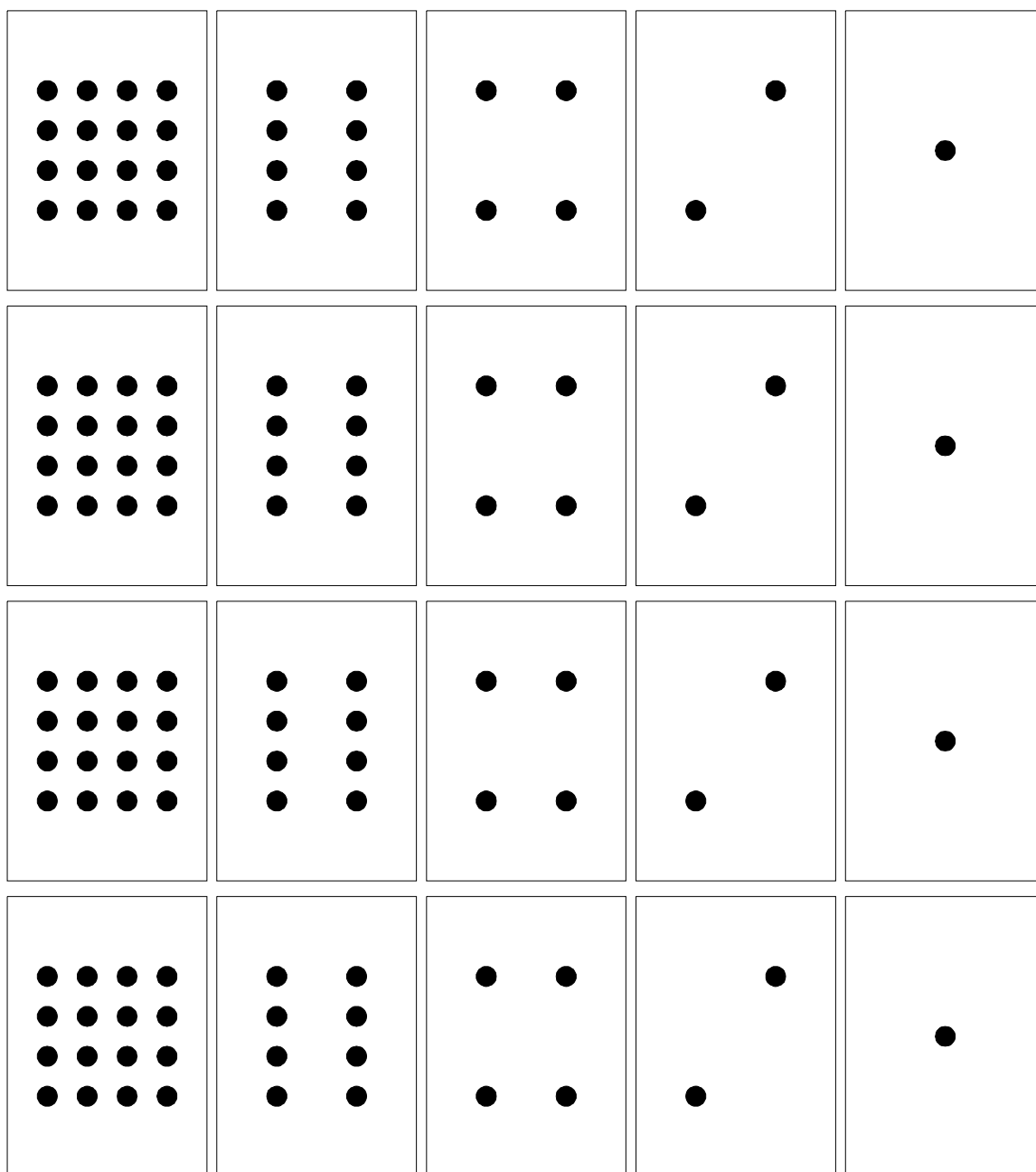


Найдите способ получить 3, 12, 19. Существует ли еще способы получения этих чисел? Какое самое большое число вы можете получить? А самое маленькое число? Какие числа вы можете получить в промежутке между самым большим и самым маленьким числами?

**Дополнительно для любознательных:** Попробуйте получить числа в следующем порядке 1, 2, 3, 4 . Можете ли вы увидеть закономерность в переворачивании карточек при увеличении числа на единицу?

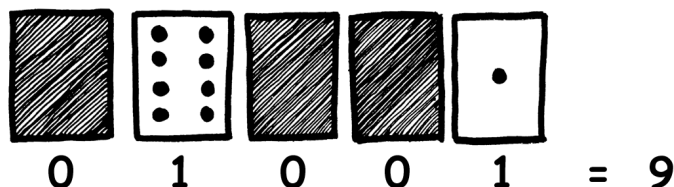
## Фотокопия для учителя: Двоичные числа

---



## Лист деятельности: Работа с двоичными числами

В двоичной системе используются ноль и единица как происходящее событие с переворотом карточек вверх или вниз. **0** указывает, что карточка без точек, и **1** – с изображением точек. Например:



Вы можете указать десятичное число равное **10101**? А **11111**? В какой день месяца вы родились? Запишите это в двоичной системе. Укажите в двоичной системе день рождения вашего друга.

**Поработайте с этими закодированными числами:**

$$\begin{matrix} \boxed{\times} & \boxed{\checkmark} & \boxed{\times} & \boxed{\times} & \boxed{\checkmark} \\ (\checkmark=1, \times=0) \end{matrix} =$$

$$\begin{matrix} \uparrow & \downarrow & \uparrow \\ (\uparrow=1, \downarrow=0) \end{matrix} =$$

$$\begin{matrix} \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ (\odot=1, \circ=0) \end{matrix} =$$

$$\begin{matrix} \text{box up} & \text{box down} \\ (\text{box up}=1, \text{box down}=0) \end{matrix} =$$

$$\begin{matrix} \text{happy face} & \text{sad face} \\ (\text{happy face}=1, \text{sad face}=0) \end{matrix} =$$

$$\begin{matrix} \text{thumbs up} & \text{thumbs down} & \text{thumbs up} & \text{thumbs down} \\ (\text{thumbs up}=1, \text{thumbs down}=0) \end{matrix} =$$

$$\begin{matrix} + & + & \times & + \\ (+=1, \times=0) \end{matrix} =$$

$$\begin{matrix} \cup & \cup & \cup & \cup & \cup \\ (\cup=1, \cap=0) \end{matrix} =$$

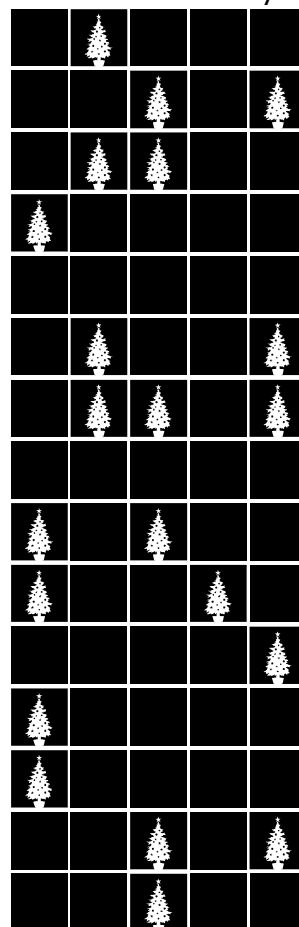
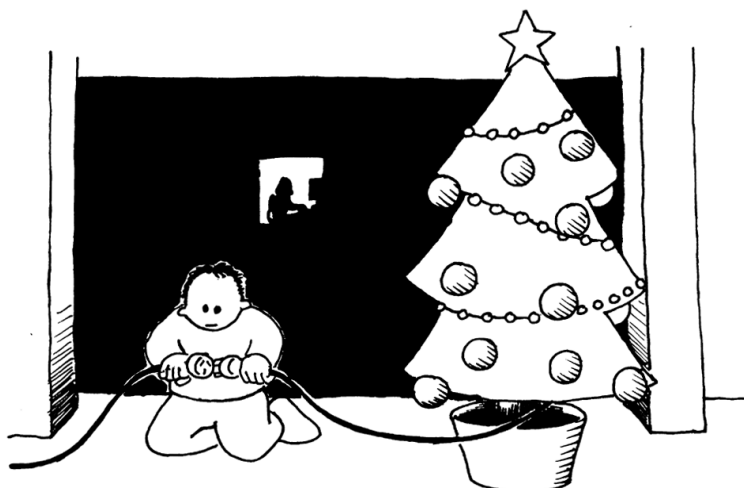
$$\begin{matrix} \blacktriangle & \blacktriangledown & \blacktriangle & \blacktriangledown & \blacktriangledown \\ (\blacktriangle=1, \blacktriangledown=0) \end{matrix} =$$

$$\begin{matrix} \spadesuit & \spadesuit & \spadesuit & \spadesuit & \spadesuit \\ (\spadesuit=1, \clubsuit=0) \end{matrix} =$$

**Дополнительно для любознательных:** С помощью набора стержней длины 1, 2, 4, 8 и 16 (единиц измерения) вы можете собрать стержни длины до 31 той же единицы измерения. Или, вы очень удивите взрослых, если покажите им, как можно взвесить тяжелые вещи, такие как чемоданы и коробки, имея несколько весов и используя тот же принцип, что и со стержнями!

## Лист деятельности: Отправление секретных сообщений

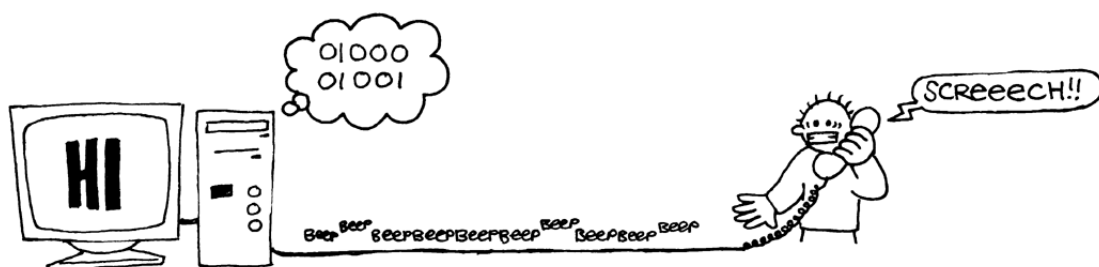
Том случайно закрыли на верхнем этаже универмага. Это, как раз произошло перед Рождеством, и он очень хотел выбраться домой со своими подарками. Как ему это сделать? Он пытался дозвониться, даже кричал, но никто не отвечал. На противоположной стороне улицы он увидел в окне девушку, работающую на компьютере поздней ночью. Как привлечь её внимание? Том огляделся вокруг, чтобы ему использовать? Его осенила блестящая идея, ведь он может использовать ёлочные гирлянды, чтобы отправить сообщение! Он находит все лампочки и розетки и подключает так, что можно сигнализировать: включить или выключить. Том использует простой двоичный код, который девушка, на другой стороне улицы, несомненно, понимает. А вы так сможете?



1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z

## Лист деятельности: Электронная почта и Модемы

Компьютеры, подключенные к интернету через модем, также используют двоичную систему для отправления сообщений. Отличие только в том, что они используют звуковые сигналы. Высокий пронзительный звуковой сигнал используется для передачи единицы, а низкий сигнал – для нуля. Эти сигналы передаются так часто, что мы можем услышать только ужасный непрерывный жужжащий звук. Если вы никогда не слышали, как работают модемы, подключенные к интернету, то попробуйте позвонить по факсу, которые также использует модем для передачи информации.



Используя тот же код, что и Том использовал в универмаге, попробуйте отправить сообщение вашему другу по электронной почте. Сделать это просто, хотя это будет не так быстро как по реальному модему!



## Лист деятельности: Подсчёт чисел больше, чем 31

Посмотрите опять на карточки с точками. Если бы вы делали следующую карточку, то, сколько точек было бы на ней изображено? А как насчет ещё одной после предыдущей карточки? Какое правило необходимо соблюдать, чтобы сделать новые карточки? Как вы можете увидеть, достаточно только несколько карточек, чтобы посчитать очень большие числа.

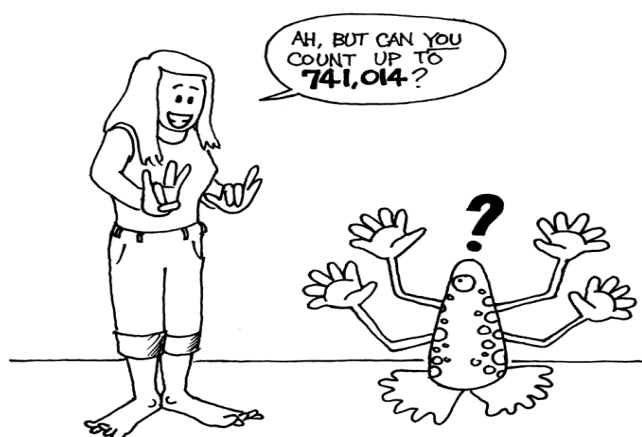
Если вы посмотрите на последовательность внимательно, то увидите интересные соотношения: **1, 2, 4, 8, 16...**

Попробуйте сложить:  $1 + 2 + 4 = ?$  К чему это приведет? Сейчас попробуйте  $1 + 2 + 4 + 8 = ?$  Что произойдет, если вы сложите все числа?

Вы когда-нибудь слышали: "пальцы прогуливаются"? Вот сейчас вы можете вашим пальцам позволить посчитать, но вы можете получить числа больше, чем десять – нет, если вы конечно не инопланетянин. При использовании двоичной системы, пусть каждый палец представляет одну из карточек с точками и у вас есть возможность посчитать от 0 до 31. Это 32 числа. (Не забываете, что 0 тоже число!)

Попробуйте посчитать по порядку с помощью пальчиков. Если палец вверх, то это означает один, если вниз – то ноль. Вы можете получить числа от 0-до 1023, если будете использовать обе руки! Это 1024 числа!

Если вы сможете сгибать и пальцы ног (теперь вы - точно инопланетянин), то у вас будет возможность посчитать больше чисел. Если с помощью одной руки, вы можете посчитать 32 числа, а двумя руками можно посчитать  $32 \times 32 = 1024$  чисел, какого наибольшего числа могут достичь Мисс Гибкие-Пальчики?





## Лист деятельности: Подробнее о двоичных числах

1. Ещё одно интересное свойство двоичных чисел, то, что происходит, когда ноль ставят справа. Если основание системы счисления - 10 (десятичная система), прибавление нуля справа означает умножение на 10. Например, 9 получаем 90, 30 получаем 300.
2. Но что происходит, когда ноль ставят справа в двоичной системе? Попробуйте это:

$$\begin{array}{ccc} 1001 & \rightarrow & 10010 \\ (9) & & (?) \end{array}$$

Приведите несколько других примеров для проверки вашей гипотезы. Что это за правило? Как вы думаете, почему это происходит?

Каждая из карточек использовалась нами, для представления одного «бит» информации на компьютере («бит» сокращенное от 'binary digit'). Таким образом, в нашем алфавитном коде мы использовали только пять карточек, или пять «бит». Однако компьютер должен понимать - эта буква прописная или заглавная, а также распознавать цифры, знаки препинания и специальные символы, такие как, \$ или ~.

Подойдите и посмотрите на клавиатуру и убедитесь, как много символов компьютер должен распознавать. Итак, сколько бит надо компьютеру для хранения всех символов?

Большинство компьютеров сегодня используют представление, называемое ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange Американский стандартный код для обмена информацией), который основывается на использовании этого числа битов для представления символов, но некоторые не англо-говорящие страны вынуждены использовать более длинные коды.

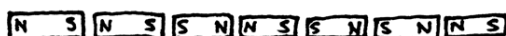


# О чём это всё?

Компьютеры используют двоичную систему для представления информации. Она называется двоичной, потому что используются только два числа (люди обычно используют систему с основанием 10). Каждый ноль или единица называется бит (**binary digit** двоичная цифра). Бит, как правило, представлен в ячейке памяти компьютера транзистором, который включен или выключен, или конденсатором, который заряжен или разряжен.



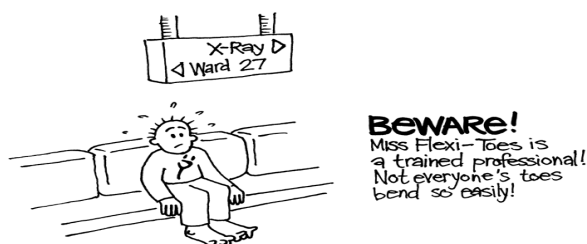
Если данные должны быть переданы по телефонной линии или радиоканалу, высокие и низкие тона, используются, соответственно, для единиц и нулей. На магнитных дисках (жестких дисках или дискетах) и лентах, биты, представлены направлением магнитного поля на покрытии поверхности, либо Север-Юг или Юг-Север.



На Audio CD, CD-Rom, DVD биты хранятся оптически – часть поверхности либо отражает свет или не отражает свет, соответственно.



Один бит не может представлять много информации, поэтому они, как правило, объединены в группы по восемь, которые могут представлять числа от 0 до 255. Группа из восьми битов называется байтом. Скорость компьютера зависит от того количества битов, которые он может обработать одновременно. Например, 32-битный компьютер может обработать 32-битные числа за одну операцию, в то время как 16-битный компьютер, должен прерываться для разбиения 32-битных чисел на мелкие части, что замедляет его скорость. Итак, биты и байты это всё, что использует компьютер, чтобы хранить и предавать числа, текст и другую информацию. При выполнении другого вида деятельности, мы увидим, как другие виды информации могут быть представлены на компьютере.



# Решения и подсказки

---

## Двоичные числа (стр. 5)

3 необходимы карточки с 2 и 1  
12 необходимы карточки с 8 и 4  
19 необходимы карточки с 16, 2 и 1

Существует только единственный способ представления этих чисел.

Наибольшее число, которое вы можете получить - это 31. Наименьшее - это 0. Вы можете получить любое число между этими двумя числами, и каждое представление этого числа единственно.

**Любознательным:** Увеличение на одно число - это последовательный переворот всех карточек справа налево, пока на всех карточках не будут изображены точки.

## Работа с двоичными числами (стр. 7)

$10101 = 21$ ,  $11111 = 31$

## Отправление секретных сообщений (стр. 8)

Кодовое сообщение: HELP IM TRAPPED (ПОМОГИТЕ Я В ЛОВУШКЕ)

## Подсчёт чисел больше, чем 31 (стр. 10)

Если вы будете прибавлять числа, сумма всегда будет на единицу меньше, чем следующее число в последовательности.

Мисс Гибкие-пальчики смогут посчитать  $1024 \times 1024 = 1,048,576$  чисел — 0 до 1,048,575!

## Подробнее о Двоичных числах (стр. 11)

Когда вы приписываете ноль к числу справа, то число удваивается.

Единица смещается дважды, т.е. на два разряда, и поэтому конечное число удваивается (В 10 системе добавления нуля справа - это умножение на 10).

Компьютеру необходимо 7 битов для хранения всех разрядов целых чисел. Это позволяет иметь до 128-бит целых чисел. Обычно 7 битов хранятся в байте 8-бит, с одним битом пустым.

# Деятельность 2

---

## Цвет Чисел— *Представление изображения*

### Введение

Компьютеры хранят рисунки, фотографии и другие изображения, используя только цифры. Данный вид деятельности демонстрирует, как они это делают.

### Учебные направления

- ✓ Математика: Геометрия от 2 уровня. Изучение формы и пространства.

### Навыки

- ✓ Вычисления
- ✓ График

### Возраст

- ✓ От 7 лет

### Материалы

- ✓ Сделать прозрачные пленки ОНР «ОНР Учителя»: Цвет числа (стр. 16)

Для каждого ребенка необходимо:

- ✓ Лист деятельности: Детский факс (стр. 17)
- ✓ Лист деятельности: Сделайте свой рисунок (стр. 18)

# Цвет Чисел

---

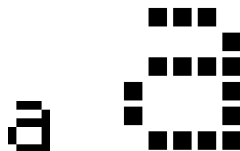
## Указания

### Вопросы для обсуждения

1. Как сделать факсимильную связь (факс)?
2. В каких ситуациях компьютеру необходимо хранить рисунки? (Программа для рисования, игры с графикой, или мультимедийные системы.)
3. Как компьютер может хранить рисунки, когда он использует только цифры?

(Вы можете показать детям, как отправить и/или получить факс при подготовке к данному виду деятельности).

### Показ прозрачной пленки ОНР



Компьютерные экраны делятся на сетки мелких точек, называемых пикселями (**picture elements** элементы изображения).

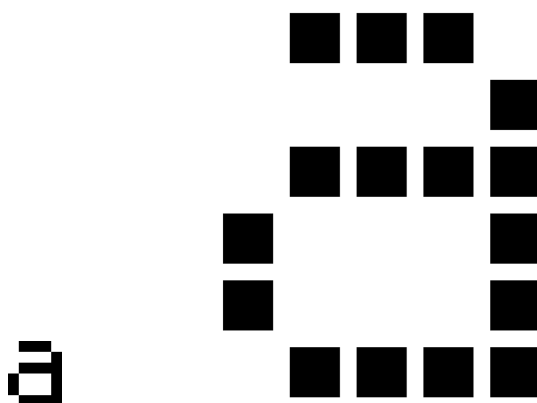
На черно-белом изображении, каждый пиксель черный или белый.

Буква “а” увеличена для показа пикселей. Когда компьютер хранит изображение, всё что необходимо ему знать – это где чёрная точка, а где белая.

					1, 3, 1
					4, 1
					1, 4
					0, 1, 3, 1
					0, 1, 3, 1
					1, 4

Этот рисунок показывает нам, как изображение может быть представлено в виде чисел. Первая строка состоит из одного белого пикселя, трех чёрных, и затем ещё одного белого. Таким образом, первая строка представлена в виде 1, 3, 1. Первое число всегда показывает количество белых пикселей. Если же надо показать первым, чёрный пиксель, то начинают с нуля. На листе (стр. 17) приведены рисунки, которые вы можете использовать для показа этого метода детям.

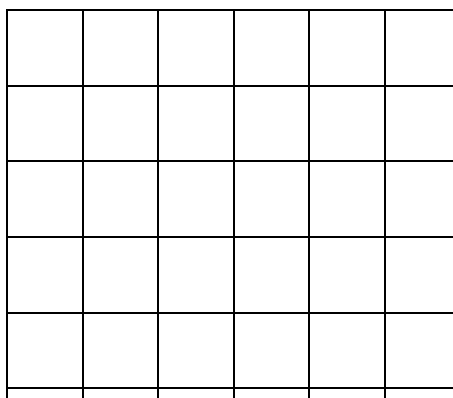
## ОНР Учителя: Цвет чисел



▲ Эта буква "а" - с экрана компьютера, показана в увеличенном виде для демонстрации пикселей, из которых состоит изображение.

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

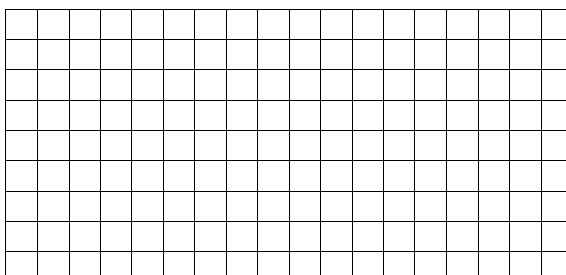
▲ Тоже изображение кодируется с использованием чисел.



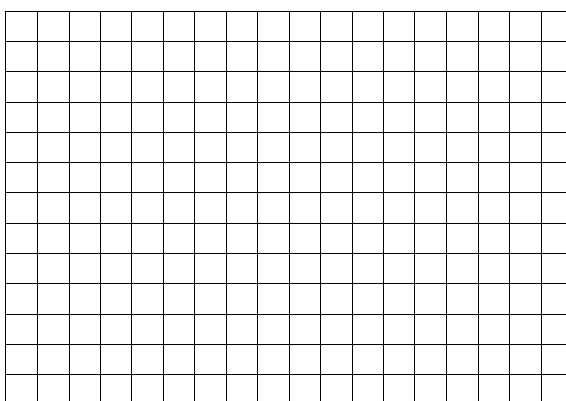
▲ Сетка-шаблон (для учебных целей)

## Лист деятельности: Детский факс

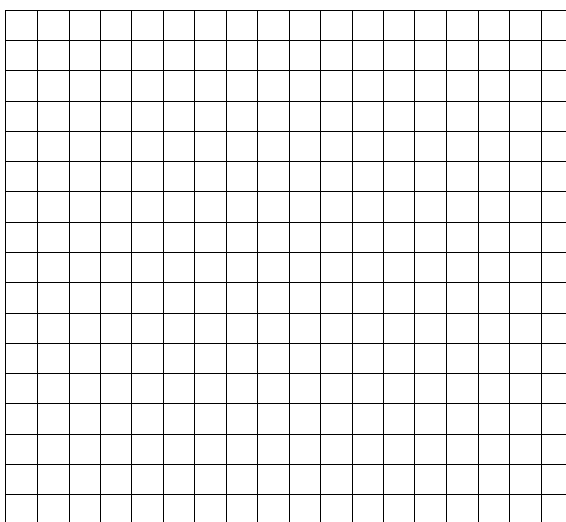
Первый рисунок самый простой, а последний является более сложным. Легко допустить ошибку, поэтому лучше использовать цветной карандаш и ластик, для удобства!



4, 11  
4, 9, 2, 1  
4, 9, 2, 1  
4, 11  
4, 9  
4, 9  
5, 7  
0, 17  
1, 15



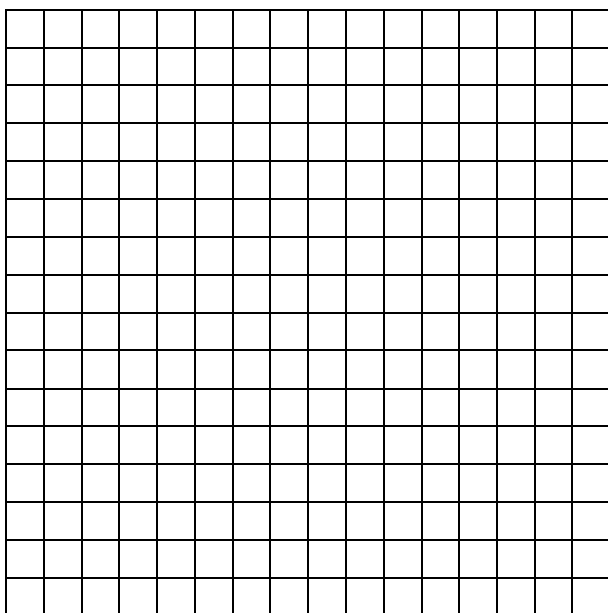
6, 5, 2, 3  
4, 2, 5, 2, 3, 1  
3, 1, 9, 1, 2, 1  
3, 1, 9, 1, 1, 1  
2, 1, 11, 1  
2, 1, 10, 2  
2, 1, 9, 1, 1, 1  
2, 1, 8, 1, 2, 1  
2, 1, 7, 1, 3, 1  
1, 1, 1, 1, 4, 2, 3, 1  
0, 1, 2, 1, 2, 2, 5, 1  
0, 1, 3, 2, 5, 2  
1, 3, 2, 5



6, 2, 2, 2  
5, 1, 2, 2, 2, 1  
6, 6  
4, 2, 6, 2  
3, 1, 10, 1  
2, 1, 12, 1  
2, 1, 3, 1, 4, 1, 3, 1  
1, 2, 12, 2  
0, 1, 16, 1  
0, 1, 6, 1, 2, 1, 6, 1  
0, 1, 7, 2, 7, 1  
1, 1, 14, 1  
2, 1, 12, 1  
2, 1, 5, 2, 5, 1  
3, 1, 10, 1  
4, 2, 6, 2  
6, 6

## Лист деятельности: Сделайте свой рисунок

Сейчас, когда вы знаете, как числа представляют рисунки, почему бы вам не сделать своё собственное закодированное изображение, для друга? Нарисуйте свою картинку на верхней сетке, а когда вы закончите, напишите код с номерами на нижней сетке. Разрежьте по пунктирной линии и отдайте нижнюю сетку другу, для раскраски. (Примечание: вы можете не использовать полностью всю сетку, если не хотите — просто оставьте несколько пустых строк в нижней части сетки, если ваш рисунок не заполняет всю сетку).



---

---

---

---

---

---

---

---

---

---

---

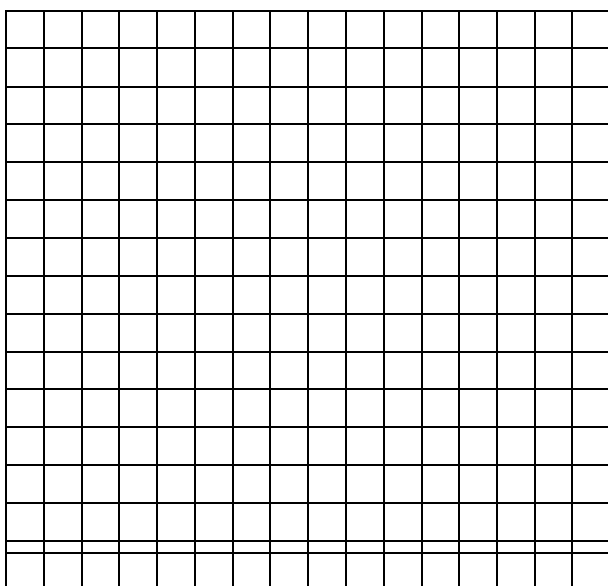
---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

---

---

---

---

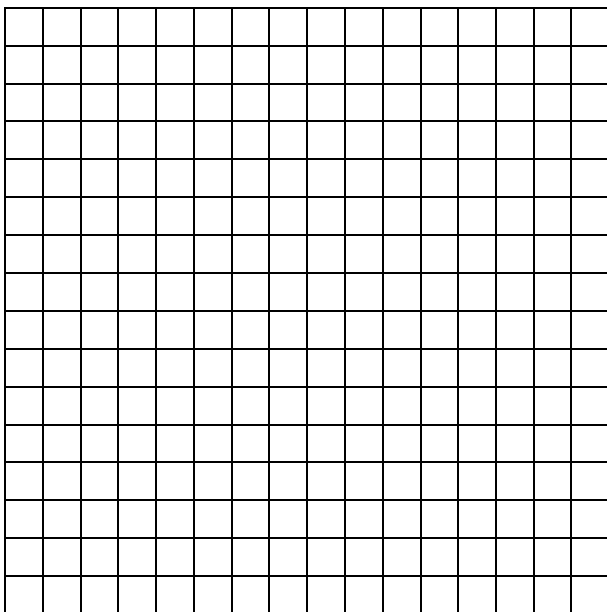
---

---



## Лист деятельности: Сделайте свой рисунок

**Дополнительно для любознательных:** Если вы хотите создать цветной рисунок, то можете использовать число для указания цвета (например, 0 - чёрный, 1 - красный, 2 – зелёный и т.д.) Тогда уже два числа будут использованы для рисования пикселей: первое – показывает длину, а второе – цвет. Попробуйте сделать цветное изображение для друга. Не забудьте подсказать другу, какое число и для какого цвета вы придумали!



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

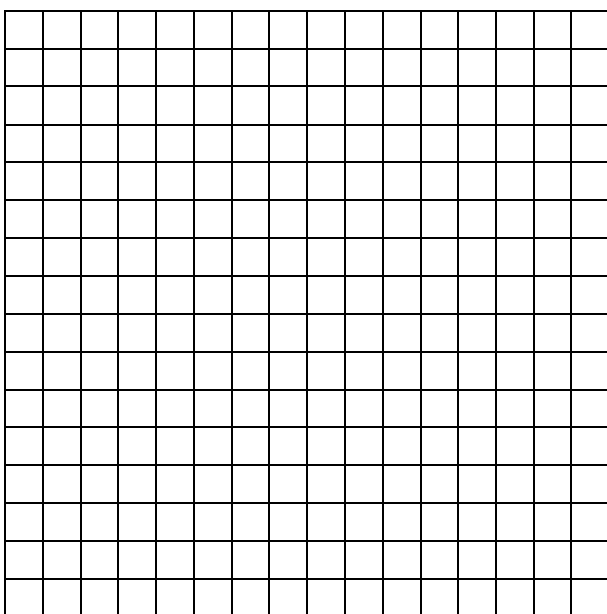
---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

### **Вариации и расширения**

1. Попробуйте перенести рисунок с кальки на верхнюю часть сетки, так чтобы окончательное изображение можно было просматривать без сетки. Изображение тогда будет более чётким.
2. Вместо раскраски сетки, дети могут использовать квадраты липкой бумаги, или другие объекты на магнитной доске.

### **Вопрос для обсуждения**

Существуют, как правило, ограничения на длину пикселей, поскольку длина представлена двоичным числом. Как бы вы представили двенадцать чёрных пикселей, если можно использовать числа только до семи? (Хороший способ кодирования заключается в следующем: сначала представить семь чёрных пикселей, далее, нулевой белый пиксель, и затем, ещё пять чёрных.)

# О чём это всё?

---

Факсимильный аппарат, на самом деле, простой компьютер, который сканирует черные и белые пиксели, с одной страницы размером  $1000 \times 2000$  пикселей, и передает, с использованием модема, на другой факсимильный аппарат, который уже изображает эти пиксели, на соответствующей странице. Часто изображения факса имеют большие блоки белого (например, широкие поля), или черного пикселей (например, горизонтальные линии). В цвете рисунков содержится много повторяющихся элементов. Чтобы сэкономить на объеме дискового пространства, необходимого для других изображений, программисты используют метод сжатия. Метод, который используется в этом виде деятельности, называется 'групповое кодирование' и является эффективным способом сжатия изображений. Если бы мы не сжимали изображения, то передача заняла больше времени, и места - для хранения. Это бы сделало невозможным отправку факсов или размещение рисунков на веб-страницах. Например, факс - изображения сжимаются до седьмой сотой от их первоначального размера. Без сжатия они будут передаваться в семь раз дольше!

Фотографии и рисунки часто сжимаются до десятых и даже сотых от их первоначального размера (с помощью различных технологий). Это позволяет хранить намного больше изображений на диске, а это значит, что и увеличится возможность для просмотра через Интернет.

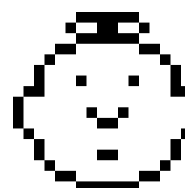
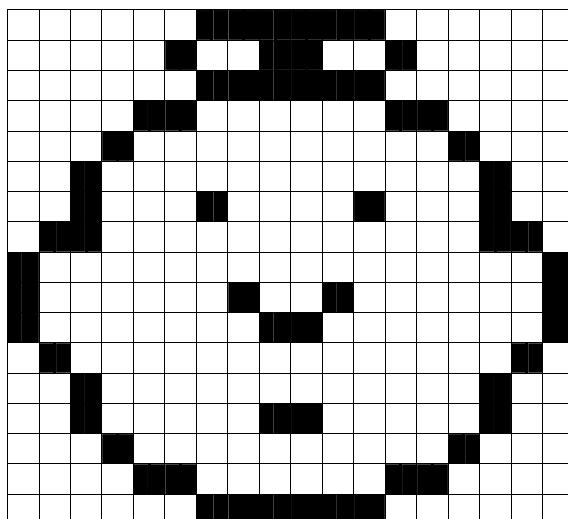
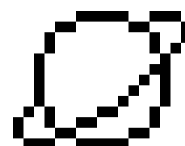
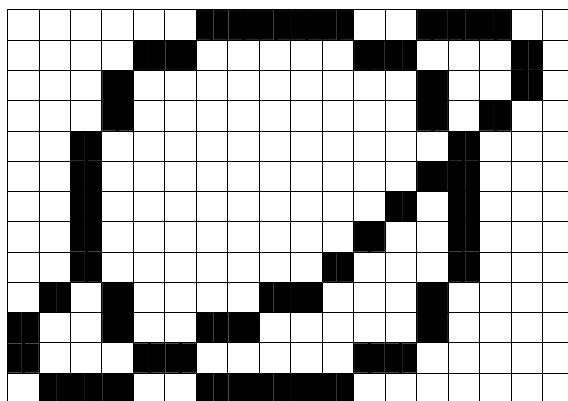
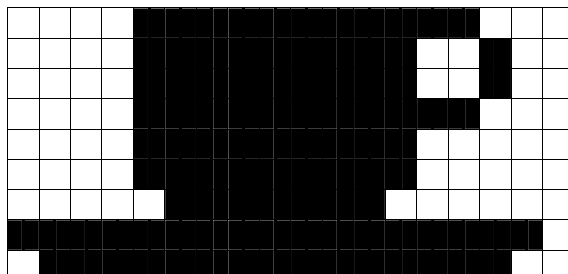
Программист имеет возможность выбрать, какой метод лучше всего подходит ему, для передачи того или иного изображения.



# Решения и подсказки

---

Ответы к Листу: Детский факс



# Деятельность 3

---

## Вы можете это повторить! —Сжатие текста

### Введение

Так как компьютеры ограничены возможностью хранения информации, то необходимо представить информацию настолько эффективно, насколько это допустимо. Этот процесс называют сжатием. Кодирова данные до их сохранения, и декодируя после их восстановления, компьютер может хранить больше данных, или отправлять их быстрее посредством Интернета.

### Учебные направления

- ✓ Английский язык: Распознавание закономерности в словах и в тексте.
- ✓ Технология: Технологические знания и понимания. Как работают компьютеры.

### Навыки

- ✓ Повторение написанного текста.

### Возраст

- ✓ От 9 и старше

### Материалы

ОНР плакат, сделанный из пленки ОНР, для Учителя: Вы сможете это повторить! (стр. 25)

Каждому ребёнку необходимо:

- ✓ Лист деятельности: Вы сможете это повторить! (стр. 26)
- ✓ Лист деятельности: Для любознательных (стр. 27)
- ✓ Лист деятельности: Коротко и ясно (стр. 28)
- ✓ Лист деятельности: Для очень любознательных (стр. 29)

# Вы сможете это повторить!

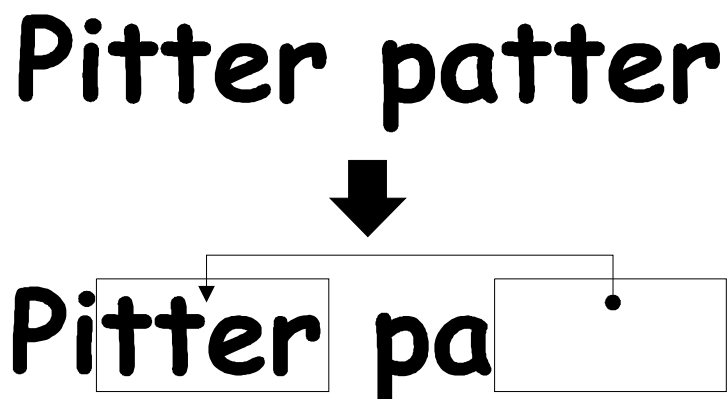
---

## Указания

Компьютеры должны хранить и передавать большой объем данных. Но так как они не могут использовать для хранения данных или для передачи больших массивов информации через сети связи, неограниченное количество ячеек памяти, то они сжимают данные и информацию, и выглядит это следующим образом.

## Демонстрация и обсуждение

Покажите «Дождь» ОНР (стр. 25). Найдите одинаковые словосочетания (шаблоны) в этом стихотворении. Сможете ли вы найти группы из 2 или более символов, которые повторяются, а может быть целые слова или фразы? (Замените их полями как показано ниже на диаграмме).



## The Rain

---

Pitter patter

Pitter patter

Listen to the rain

Pitter patter

Pitter patter

On the window pane

## Лист деятельности: Вы сможете это повторить!

Много слов и символов отсутствуют в этом стихотворении. Сможете ли вы заполнить недостающие символы и слова, чтобы получить правильные предложения? Вы найдете их в полях, на которые указывает стрелка.

Peas por idg h. t, cold, in the p, Nine days, Some like t, , , , .

Peas por idg h. t, cold, in the p, Nine days, Some like t, , , , .

Сейчас выберете простое стихотворение или ритмичную скороговорку и спроектируйте свою задачу. Удостоверьтесь, что все ваши стрелки указывают на начальную часть текста. Ваше стихотворение должно быть декодировано слева направо, и сверху донизу, таким образом, как мы всегда читаем текст.

**Сложная задача:** Обратите внимание, что некоторые оригинальные слова необходимо сохранить!

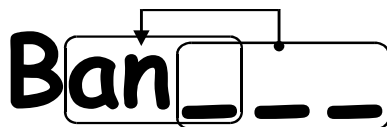
Вот некоторые из них: Three Blind Mice (Три самоуверенных мышонка), Mary Mary Quite Contrary (Мэри Мэри Абсолютное Упрямство), Hickory Dickory Dock (Хикори Дикори Док) — или другие герои книг Dr. Seuss (др. Сеусса)!

**Подсказка:** Попытайтесь избежать нагромождения стрелок. Оставьте больше пространства около предложений и слов, для указания полей и записи в них шаблонов. Проще проектировать задачу, если вы вначале записываете стихотворение, а затем решаете, где у вас будет поле.



## Лист деятельности: Для любознательных

Как вы будете решать эту задачку?



Иногда недостающий текст указывает на часть себя. В этом случае, чтобы декодировать правильно, необходимо все символы повторить слева направо. Тогда каждый символ можно будет повторить прежде, чем он будет востребован. Это используется в компьютерах, когда он кодирует длинный специфический символ или шаблон.

Попытайтесь придумать свой пример.

В компьютерах поля и стрелки представляются числами. Например,

**Banana**

можно записать, как **Ban(2,3)**. "2" указывает, что необходимо читать с символа, записанного на два символа раньше, чтобы найти точку начала для повторения,

**Ban\_ \_ \_**

и "3" указывает на необходимость повторения трёх последовательных символа:

**Bana\_ \_**

**Ban<sup>an</sup>\_**

**Ban<sup>ana</sup>**



Так как используются два числа, чтобы закодировать это слово, то тогда необходимо сжимать и группы из этих двух или более чисел, иначе нет никакого смысла в таком сохранении пространства. Фактически, размер файла увеличивается, если использовать два числа для кодирования одного символа.

Составьте некоторые слова, закодированные таким же способом, для компьютера, как если бы они были сжаты им. Ваши друзья смогут их декодировать?

## Лист деятельности: Коротко и ясно

**Как много слов вам необходимо в этом примере?** Представьте себе, что вы - компьютер, пытающийся вместить на свой диск столько информации, сколько это возможно. Вычеркните все группы двух и более символов, которые уже встречались. Здесь они такие длинные уже не нужны, так как могут быть заменены указателем. Ваша цель состоит в том, чтобы получить как можно больше вычеркнутых символов, настолько, насколько это возможно.

I know an old lady who swallowed a bird  
How absurd! She swallowed a bird!  
She swallowed the bird to catch the  
spider  
That wriggled and jiggled  
and tickled inside her  
She swallowed the spider to catch the fly  
I don't know why she swallowed a fly  
Perhaps she'll die...

## Лист деятельности: Для очень любознательных

### Готовы к варианту, действительно жёсткого сжатия?

Нижеприведенный текст, с помощью компьютерной программы был сжат, и вычеркнуты, по крайней мере, 1 633 символа. А сколько вы можете найти? Помните, только группы двух или большего количества повторяемых символов могут быть выброшены. Удачи!

Once upon a time, long, long ago, three little pigs set out to make their fortunes. The first little pig wasn't very clever, and decided to build his house out of straw, because it was cheap. The second little pig wasn't very clever either, and decided to build his house out of sticks, for the "natural" look that was so very much in fashion, even in those days. The third little pig was much smarter than his two brothers, and bought a load of bricks in a nearby town, with which to construct a sturdy but comfortable country home.

Not long after his housewarming party, the first little pig was curled up in a chair reading a book, when there came a knock at the door. It was the big bad wolf, naturally.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the first little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he *did* huff, and he *did* puff, and the house soon collapsed. The first little pig ran as fast as he could to the house of sticks, and was soon safe inside. But it wasn't long before the wolf came calling again.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the second little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he *did* huff, and he *did* puff, and the house was soon so much firewood. The two terrified little pigs ran all the way to their brother's brick house, but the wolf was hot on their heels, and soon he was on the doorstep.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the third little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he huffed, and he puffed, and he huffed some more, but of course, the house was built of brick, and the wolf was soon out of breath. Then he had an idea. The chimney! He clambered up a handy oak tree onto the roof, only to find that there *was* no chimney, because the third little pig, being conscious of the environment, had installed electric heating. In his frustration, the wolf slipped and fell off the roof, breaking his left leg, and severely injuring his pride. As he limped away, the pigs laughed, and remarked how much more sensible it was to live in the city, where the only wolves were in the zoo. And so that is what they did, and of course they all lived happily ever after.

# О чём это всё?

---

Емкость запоминающего устройства компьютеров растет с невероятной скоростью – за 25 лет, объём памяти на типовом компьютере вырос в миллион раз, но мы всё равно хотим хранить все больше и больше информации. Компьютеры могут хранить целые книги или даже библиотеки, и теперь и музыку и кинофильмы также, если достаточно памяти для этого. Большие файлы – проблема Интернета, так как им необходимо много времени для загрузки. Мы также пытаемся сделать компьютеры меньших размеров – даже в сотовых или наручных часах, вы ожидаете, что такие компьютеры будут хранить много информации! Но решение этой проблемы есть. Вместо того, чтобы покупать больше пространства памяти, или более быстрое сетевое подключение, мы можем *сжать* данные так, чтобы они заняли меньше места. Этот процесс упаковки и распаковки данных компьютером производится автоматически. Всё, на что мы могли бы обратить внимание – то, что пространство диска вмещает больше данных, или, что web-страница отображает быстрее, т.е. компьютер фактически больше обрабатывает информации. Были изобретены различные методы сжатия. Метод, используемый здесь нами, основанный на принципе, чтобы указывать на более раннее вхождение участков памяти текста, часто упоминается как ‘Ziv-Lempel coding,’ или ‘LZ coding’, изобретенный двумя израильскими профессорами в 1970-ых. Этот метод может использоваться для любого языка и может легко быть разделен на два размера сжимаемых данных. Этот метод иногда упоминается на персональных компьютерах как ‘zip’, а также используется для изображений формата ‘GIF’ (формат обмена графическими данными), также и в высокоскоростных модемах. В случае модемов, этот метод уменьшает количество данных, которые должны быть переданы по телефонной линии, тем самым процесс идет быстрее. Один из методов, основан на идее, что у символов, которые используются чаще, должны быть более короткие коды. Азбука Морзе использует эту идею.

## Решения и подсказки

---

**Вы сможете это повторить!** (стр. 26)

**Pease porridge hot,  
Pease porridge cold,  
Pease porridge in the pot,  
Nine days old.**

**Some like it hot,  
Some like it cold,  
Some like it in the pot,  
Nine days old.**

# Деятельность 4

---

## Фокус с карточками – Обнаружение ошибки & Исправление

### Краткое содержание

Когда данные хранятся на диске или переданы с одного компьютера на другой, то мы обычно предполагаем, что они не изменяются при этом. Но иногда некоторые процессы проходят не так как надо, и данные изменяются случайно. Мы будем использовать фокус, чтобы показать, как обнаружить повреждение данных и исправить.

### Учебные направления

- ✓ Математика: Числа Уровень 3 и выше. Исследование и оценка вычислений.
- ✓ Алгебра: Уровень 3 и выше. Исследование моделей и отношений.

### Навыки

- ✓ Счет
- ✓ Распознавание нечётных и чётных чисел

### Возраст

- ✓ 9 лет и старше

### Материалы

- ✓ Ряд 36 “плат” магнитных карточек, окрашенных только с одной стороны
- ✓ Металлическая доска (белая ) для демонстрации.

Для каждой пары детей необходимо:

- ✓ 36 одинаковых карточек, окрашенных только с одной стороны.

# “Фокус”

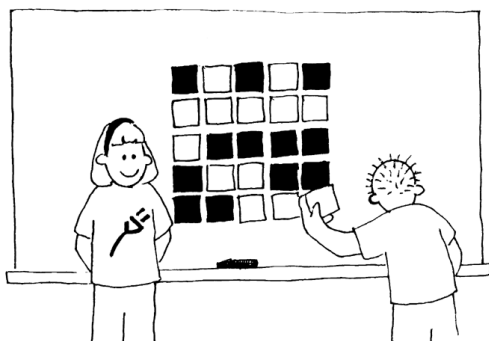
---

## Демонстрация

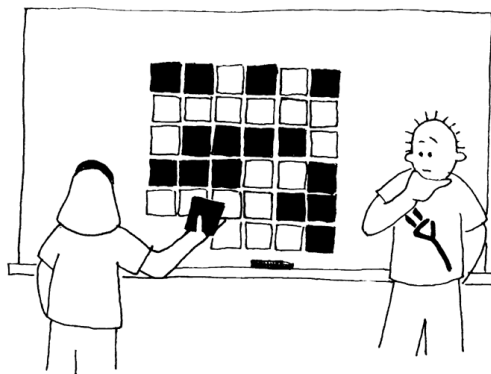
Вот вы перевоплощаетесь в фокусника!

Вам нужен комплект одинаковых, двухсторонних карточек. (Вырежете из большого листа карточки, окрашенные с одной стороны). Для демонстрации, легче использовать плоские магнитные карточки, с разным цветом обратных сторон — холодильные магниты идеально подходят.

1. Выберите ребенка, который расположит карточки со случайным выбором сторон, квадратом площадью  $5 \times 5$ .



Небрежно добавьте карточки в строку и столбец, “только сделать это немного сложнее”.



Эти карточки – ключи к фокусу. Вы должны выбрать цвет дополнительных карточек таким образом, чтобы они указывали на наличие чётного числа карт в каждой строке и столбце.

2. Попросите ребенка перевернуть одну карточку, при этом вы закройте глаза. Строки и столбцы, содержащие перевернутую карту, теперь будут иметь нечётное число карт одного цвета, и таким образом определится перевернутая карта.

Может быть, дети догадаются, как был осуществлён фокус?

### Научите детей фокусу:

1. Дети работают в парах, расположите перед собой карточки  $5 \times 5$ .
2. Сколько закрашенных карточек расположены в каждой строке и столбце? Их число чётное или нечётное? Помните, что 0 есть число чётное.
3. Теперь добавьте шестую карту в каждой строке справа, убедившись, что данное изменение не повлияло на количество закрашенных карточек в данной строке. Эту дополнительную карточку называют картой «чётности».
4. Добавьте шестые карты в нижней строке основания квадрата, чтобы показать, что число карточек в каждом столбце есть число чётное.
5. Теперь переверните карточку на другую сторону. На что вы должны обратить внимание в строке и столбце? (В них увеличится число закрашенных карточек). Карты четности используются, чтобы показать вам, где была допущена ошибка.
6. Сейчас верните карточки в исходное состояние, т.е. как бы осуществите «фокус».

### Дополнительная деятельность:

1. Попробуйте использовать другие объекты. Что-нибудь, что определяется двумя соответствующими 'состояниями'. Например, вы можете использовать игровые карты, монеты (орёл или решка) или карточки с 0 или 1 (чтобы связать данную деятельность с двоичной системой счисления).
2. Что случится, если две или большее количество карточек будут перевернуты? (При этом, не всегда возможно узнать, какие конкретно карточки были перевернуты, хотя можно сказать, что произошли некоторые изменения. Вы можете ограничить количество переворотов одной из двух пар карточек. С 4 переворотами может быть так, что все карты (биты) четности, будут правильны, но ошибка не будет обнаружена).
3. Другим интересным примером является рассмотрение правой нижней карты. Если выбранная вами карта является верной для столбца сверху, то является ли эта карта верной для строки слева? (Ответ да, всегда.)
4. В этом примере с картами вы использовали, при проверке на чётность, чётное количество закрашенных карточек. Можно ли применить этот способ для проверки на нечётность? (Возможно, но правая нижняя карта определяет только свою строку и столбец, т.е., если число строк и столбцов, одновременно, либо чётные, либо нечётные. Например, для  $5 \times 9$  - схема будет выполняться и для  $4 \times 6$ , но для  $3 \times 4$  - не будет выполняться.).

## Для любознательных – пример из реальной жизни

Такая же методика распознавания ошибки используется с книжными кодами. В изданных книгах есть код с десятью цифрами, обычно расположенный на обратной стороне обложки. Десятая цифра – контрольная цифра, точно так же, как карты (биты) чётности в предыдущем примере. Это означает, что, если вы заказываете книгу, используя её ISBN (International Standard Book Number), то издатели могут проконтролировать, чтобы вы не ошиблись. Они просто смотрят на контрольную сумму. Этот способ, гарантирует то, что вы не получите не заказанную вами книгу!

Вот как определить контрольную сумму. Умножьте первую цифру на десять, вторую на девять, третью на восемь, и так далее, до девятой цифры, умноженной на два. Сложите полученные значения. Например, ISBN 0-13-911991-4 определяет значение

$$\begin{aligned} & (0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) \\ & + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) \\ & = 172 \end{aligned}$$

Тогда разделите свой результат на одиннадцать. Какой остаток?

$$172 \div 11 = 15 \text{ остаток } 7$$

Если остаток – ноль, то контрольная сумма – ноль, иначе вычтите остаток из 11, чтобы получить контрольную сумму.

$$11 - 7 = 4$$

Посмотрите на обложку. Действительно ли это последняя цифра ISBN? Да! Если бы последняя цифра ISBN не была 4, то мы бы знали, что была допущена ошибка. Можно получить в результате более чем одну цифру, если контрольная сумма состояла из 10 цифр. Когда это случается, то используют символ X.



▲ Штрих-код (UPC) на коробке от Weet-Bix™

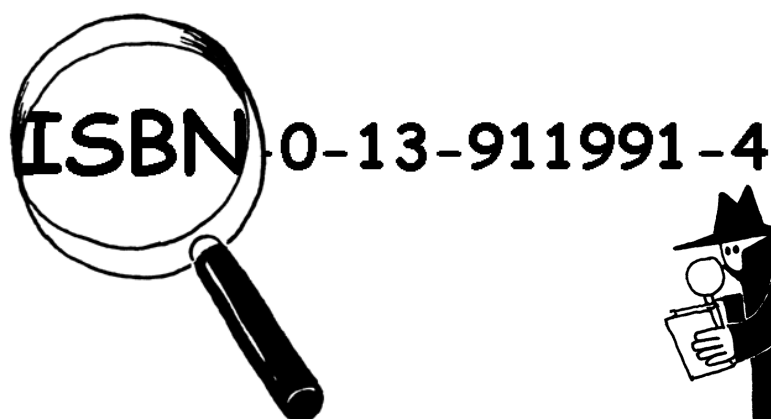
Другим примером использования контрольной цифры – штрих - коды на продуктовых товарах. Здесь используется другая формула. Если штрих-код неправильно читается, то конечная цифра отличается от своего расчётного значения. Когда это происходит, то на это указывают гудки сканера, и контролёр повторно проверяет код.



# Проверьте ту книгу!

Детективный блокбастер

Книжная компания, Inc.



Мы находим и проверяем контрольные суммы ISBN за небольшую плату.

Присоединяйтесь к нашему агентству - досмотра реальных ISBN кодов в своём классе или в библиотеке.

## Контрольные суммы всегда правильны?

Иногда бывают ошибки.

Вот некоторые из общих ошибок:

- ✗ измененное значение цифры;
- ✗ изменение в позиции двух рядом стоящих цифр;
- ✗ некоторая цифра вставлена в данное число;
- ✗ некоторая цифра удалена из данного числа

Вы можете найти книгу с символом X в контрольной сумме 10 цифр кода? Это не трудно обнаружить, так как в какой-то из 11 цифр есть одна, которая будет содержать этот символ.

Какие ошибки могут произойти, которые нельзя обнаружить? Вы сможете, изменив одну цифру, всё же получить правильную контрольную сумму? Что если две цифры поменять местами (ошибка при печати)?

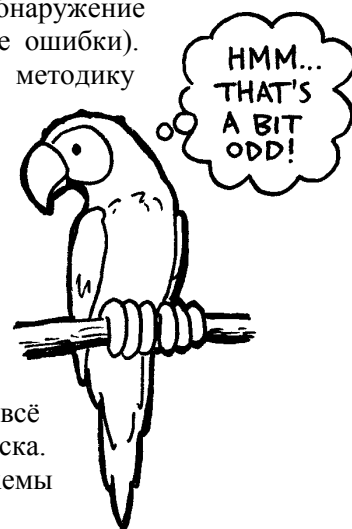
## О чём это всё?

---

Предположим, что вы вносите наличные деньги в размере \$10 на свой счет в банке. Кассир напечатал сумму вашего депозита и отправил её на центральный компьютер. Но предположим, что появляется случайная ошибка на отмеченной кассиром строке бланка депозита во время пересылки и код \$10 изменяется на \$1000. Если вы клиент, то у вас не будет никаких проблем, а для банка, ясно, что это проблема!

Важно обнаружить ошибки в переданных данных. Поэтому компьютер, при получении данных, должен удостовериться, что данные не были повреждены электрическими помехами на соответствующей строке. Иногда данные-оригиналы можно отправить повторно, после обнаружения ошибки, но в некоторых ситуациях это невыполнимо, например, если диск или лента были повреждены магнитной или электрической радиацией, подвержены тепловым или физическим воздействием. Если бы произошла ошибка, в процессе передачи данных, от далёкого космического зонда, то чрезвычайно долго бы пришлось ждать ретрансляции! (Радио - сигнал от Юпитера доходит чуть больше чем за полчаса, когда он на орбите ближайшей к Земле!)

Мы должны уметь определить повреждение данных (обнаружение ошибок) и восстановить данные – оригиналы (устранение ошибки). Используем на компьютерах ту же самую игровую методику «переворот карточек». Помещаем биты в воображаемые строки и столбцы, и добавляя биты чётности к каждой строке и столбцу, мы можем не только обнаружить ошибку, но и указать где она произошла. Определяемый искажённый бит, изменяем, и таким образом устраняем ошибку. Конечно, компьютеры используют более сложные системы контроля ошибок, способные обнаружить и исправить огромное количество ошибок. Большая часть пространства жесткого диска компьютера выделяется на исправления ошибок, для того, чтобы всё надёжно работало, даже если произойдет сбой в работе диска. Системы, используемые для этого, применяют схемы чётности.



И закончим шуткой, которая лучше ценится после выполнения этой деятельности:

**Q:** Как бы вы назвали это: «Кусков девять, кусков девять»?

**A:** Ошибка попугая.

## Решения и подсказки

---

Ошибки, которые не могут быть обнаружены, те, при которых одна цифра увеличена, а другая уменьшена. Тогда их сумма могла бы быть той же самой.

# Деятельность 5

---

## Двадцать предположений—Теория информации

### Введение

Какое количество информации содержится в 1000 страницах книги? Это больше, чем количество информации в 1000 страницах телефонной книги, или в папке незаполненных бланков на 1000 строках, или в книге «Властелин колец» Толкиена? Если мы сможем измерить это, то у нас будет возможность оценить объём необходимого пространства для хранения этой информации. Например, вы сможете прочитать следующее предложение?

Ths sntnc hs th vwls mssng.

Вы, вероятно, прочтёте это сообщение, так как гласные звуки не содержат большого количества информации. В данном виде деятельности мы знакомимся с способом измерения информационного содержания.

### Учебные направления

- ✓ Математика: Уровень 3 и выше «Числа». Исследование числа: больше, чем, меньше чем, диапазоны. Модели и последовательности.
- ✓ Английский язык

### Навыки

- ✓ Сравнение чисел и работа с диапазоном чисел
- ✓ Дедукция
- ✓ Задание вопросов

### Возраст

- ✓ От 10 и старше

### Материалы

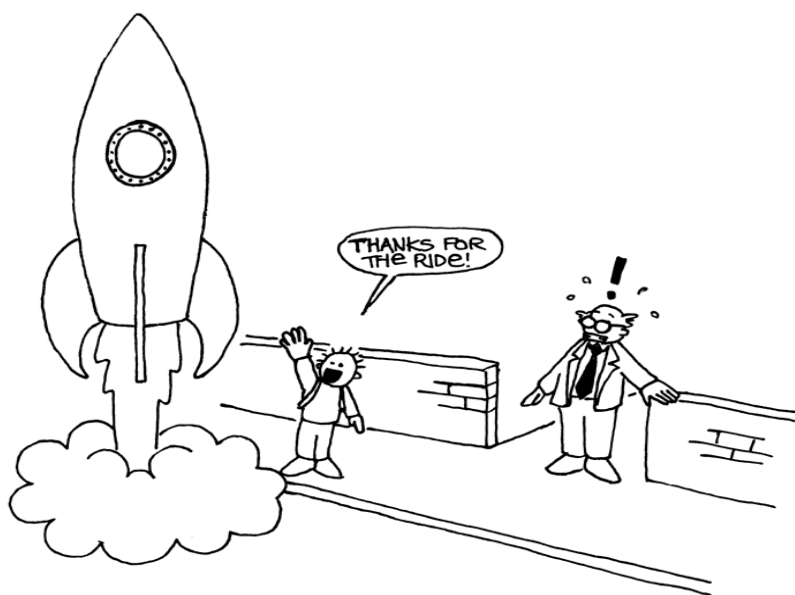
- ✓ Для основной деятельности нет необходимости в материалах
- ✓ Для расширенной деятельности каждому ребёнку необходимо:  
Лист деятельности: Деревья решений (стр. 40)

# Двадцать предположений

---

## Обсуждение

5. Обсудите с детьми понятие информации.
6. Как мы можем определить, сколько информации в книге? В книге важнее количество страниц или количество слов? Может ли в одной книге содержаться больше информации, чем в другой? Что если эта книга очень скучная, или очень интересная? Будет ли на 400 страницах книги, содержащей фразу «вздор, вздор, вздор» больше информации, чем скажем, в телефонном справочнике?
7. Объясните, что специалисты в компьютерных науках, меру информации определяют по тому, как их удивила информация (или книга!). Говоря вам кое-что, что вы уже знаете, например, когда друг, который всегда идет в школе, говорит вам, « что я иду в школу сегодня», не сообщает вам информации, потому что это не удивительно. Если же он вместо этого сказал бы, что «я сегодня прилечу в школу на вертолете», то такое сообщение удивительно, и поэтому будет содержать определенное количество информации.
8. Как можно удивительные сообщения измерить?
9. Один из методов состоит в том, чтобы измерить интенсивность угадывания информации. Если ваш друг скажет: «Угадай, как я добрался до школы сегодня», а так как вы шли вместе, то, вероятно, вы догадаетесь с первого раза. Потребовалось бы большее количество предположений прежде, чем вы бы угадали способ передвижения, например на вертолете, и еще большее количество - если бы это был космический корабль.
10. Количество информации, которую содержат сообщения, измеряется тем, насколько легко или трудно они угадываются. Следующая игра даст нам представление об этом.



# Деятельность: двадцать вопросов

---

Эта адаптированная игра - 20 вопросов. Дети могут задавать вопросы, одному из ребят, на которые можно ответить: «да» или «пока ответ не угадан». Задать можно любой вопрос при условии, что ответом будет строго: «да» или «нет».

## Предложения:

Я думаю, что:

- ✓ число между 1 и 100
  - ✓ число между 1 и 1000
  - ✓ число между 1 и 1,000,000.
  - ✓ любое целое число
  - ✓ последовательность из 6 чисел в шаблоне (для соответствующей группы).
- Угадайте, в порядке от первого до последнего (например, 2, 4, 6, 8, 10)

Подсчитайте число вопросов, которые были заданы. Это и будет количеством «информации».

## Последующее обсуждение

Какие стратегии вы использовали? Какие из них были лучшими? Покажите, что требуется только 7 предположений, для определения числа в диапазоне между 1 и 100, если вы будете делить каждый оставшийся диапазон пополам. Например:

Это число - меньше чем 50?	Да.
Это число - меньше чем 25?	Нет.
Это число - меньше чем 37?	Нет.
Это число - меньше чем 43?	Да.
Это число - меньше чем 40?	Нет.
Это число - меньше чем 41?	Нет.
Это должно быть 42!!	Да!

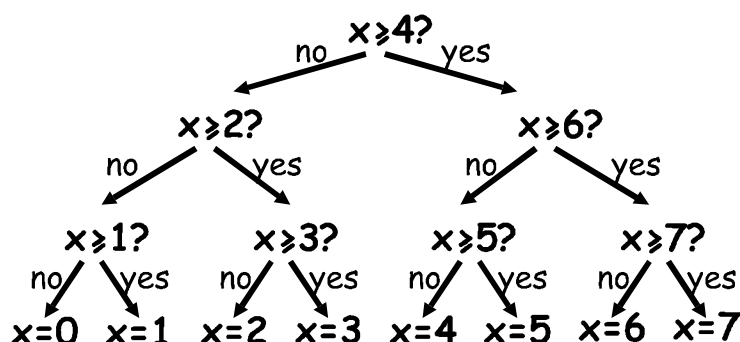
Интересно, если диапазон увеличить до 1000, то для определения задуманного числа, необходимо увеличить число вопросов не в 10 раз, а только на три вопроса. Каждый раз, при удвоении диапазона, необходим просто еще один вопрос для нахождения ответа. Полезна, в дальнейшем, будет игра для детей Mastermind.

## Для любознательных: Какое количество информации в сообщении?

IT-специалисты не только угадывают числа, но и предполагают: какой символ, более вероятно, будет следующим в слове или в предложении. Попробуйте угадывание в форме игры с короткими предложениями из 4-6 слов. Буквы должны будут угадываться в определенном порядке: с первой по последнюю. Пусть кто-нибудь записывает буквы и ведет учет всех предположений, необходимых для определения каждой буквы. Могут быть заданы любые вопросы с ответами да/нет. Например, «Это буква - т?»; «Это буква - гласная?»; «Она находится в алфавите перед буквой м?». Пробел между словами, также считается символом, и должен быть угадан. Если вы сможете обнаружить наиболее узнаваемые части сообщений, то угадывание начните именно с них.

## Лист деятельности: Деревья решений

Если вы уже знаете стратегию, как задать вопросы, то сможете передать сообщение без предварительного опроса. Вот диаграмма, называемая «деревом решений» для того чтобы определить число между 0 и 7:



Как указать путь «да/нет» в дереве решений при угадывании числа 5?

Сколько ходов «да/нет» в дереве решений вы должны сделать, чтобы определить некоторое число?

Сейчас увидим что-то очень увлекательное. В нижнем ряду дерева решений под числами 0, 1, 2, 3... запишите двоичные числа (см. деятельность 1).

Посмотрите внимательно на дерево. Если «нет» обозначим 0, а «да» – 1, что вы тогда увидите?

В игре на угадывание чисел, мы пытаемся выбрать, таким образом, вопросы, чтобы последовательность ответов было представлена одинаковым количеством ветвей дерева решений.

Создайте своё дерево решений для угадывания чисел от 0 до 15.

**Дополнительно для любознательных:** Какое «дерево» вы бы использовали для угадывания возраста кого-нибудь? Как бы вы представили «дерево» для угадывания в предложении последовательности букв?

# О чём это всё?

---

Известный американский математик (фокусник, моноциклист) Клод Шеннон провел много экспериментов с этой игрой. Он измерил объём информации в битах, где каждый ответ: да/нет - эквивалентен 1/0 бит. Он обнаружил, что количество «информации», содержащейся в сообщении зависит от того, что вам уже известно. Иногда вы можете задать один вопрос, который избавит вас от необходимости задавать ещё несколько вопросов. В этом случае сообщение содержит небольшое количество информации. Например, один бит: орёл или решка, как правило, - это информация об одном бросании монеты. Но если монета будет деформированной (неровной), и при рассмотрении переворотов: в девяти случаях из десяти - количество информации уже будет не один бит - верьте, хотите нет, а меньше. Как можно вам связать процесс бросания монеты и задания более одного вопроса, ответом на который будет: да/нет? Просто - только использовать вот такие вопросы: «при следующих двух бросках монеты выпадает орёл?» Около 80% случаев, при последовательном бросании деформированных монет, ответом будет - «да». В то время, на 20%, вам придётся ответить - «нет», при этом ещё задать два дополнительных вопроса. Но в среднем, вы будете уже задавать больше чем один вопрос по сравнению с одним бросанием монеты! Шеннон называл **информационное содержание сообщения «энтропией»**. Энтропия зависит, во-первых, не только от числа возможных исходов бросания одной монеты, но и во-вторых, также от вероятности, что это вообще произойдет. Для угадывания сообщения в случае, невероятных или удивительных событий, необходимо задать больше чем обычно вопросов, поскольку эти события содержат больше информации, по сравнению с обычным сообщением, например, в случае посадки вертолета в школе.



Энтропия сообщения очень важна для компьютерной науки. Вы можете сжать сообщение, только до её энтропии, уменьшив занимаемое этим сообщением место, и лучшие системы сжатия, подобны игре на угадывания. Поскольку компьютерная программа осуществляет «угадывание» и ответы хранятся в виде битов, то перечень вопросов может быть воспроизведен позже, при восстановлении информации. Лучшие системы сжатия могут уменьшить текстовые файлы примерно в четверть от их первоначального размера - это большая экономия памяти! Метод угадывания может быть использован для создания компьютерного интерфейса, позволяющего предсказывать, что пользователь собирается ввести в запросе! Этот метод может быть полезен людям с ограниченными возможностями, которым трудно набрать текст. Компьютер предлагает или угадывает возможные варианты запросов пользователей, а они соглашаются или нет. Хорошая система нуждается, в среднем, в двух ответах «да/нет» на символ, и может быть большим подспорьем для тех, у кого проблемы в точных движениях, необходимых для управления мышью или клавиатурой. Такие системы используются также в случаях «набора» текста в некоторых мобильных телефонах.

# Решения и подсказки

---

Ответ «да / нет» на вопрос - соответствует ровно одному биту информации, будь то простой вопрос, как: «Это число больше, чем на 50?»; или более сложный, такой, как: «Это число - между 20 и 60?»

При угадывании чисел, если вопросы выбраны определенным образом, последовательность ответов представлена только двоичными числами. Число три - «011» в двоичной системе, и оно представлено ответами «Нет, да, да». Что, то же самое, если мы запишем: «нет» - числом 0 и «да» - числом 1, в дереве решений.

Дерево можно использовать для определения возраста кого-нибудь, как смещение в сторону уменьшения значения возраста.

Решение о букве в предложении, зависит от того, какова была предыдущая буква.



# **Часть II**

**Заставим Компьютеры работать  
—Алгоритмы**

# Заставим Компьютеры работать

---

Компьютеры работают по указанному им списку инструкций. Эти инструкции позволяют компьютеру сортировать, находить и отправлять информацию. Чтобы сделать это как можно быстрее, необходимы хорошие методы нахождения информации в больших массивах данных и передачи её с помощью сети.

Алгоритм представляет собой набор инструкций для решения задачи. Понятие алгоритма занимает центральное место в компьютерных науках. Алгоритмы - это, как мы указываем компьютерам решать задачи. Некоторые алгоритмы решения быстрее, чем другие. И, многие из алгоритмов, которые были обнаружены, позволили решить задачи, требующие больших затрат времени, например, нахождения миллионного разряда десятичных знаков числа  $\pi$  (пи); или все страницы, которые содержат ваше имя на World-Wide Web; или выяснение лучшего способа упаковки пакетов в контейнер; или нахождение простых, не очень больших чисел (100-разрядов).

Слово «алгоритм» произошло от имени Мухаммед ибн Мусса Ал-Хоризм – Мухаммед, сын Моисея, из Хоризма, который известен как ученый из Дома Мудрости в Багдаде (около 800 лет). Его работы переведены с хинди на арабский, а затем и в Европе. Когда они переводились на латынь в 1200 году, то первыми словами имени были «Диксит Алгоризми» - так и говорили: «Алгоризми».

# Деятельность 6

---

## Кораблики—Алгоритмы поиска

### Введение

Компьютерам часто требуется найти информацию в больших массивах данных. Они должны быть для этого быстрым и эффективным. В этой деятельности демонстрируется три различных метода поиска: линейный поиск, бинарный поиск и хеширование.

### Учебные направления

- ✓ Математика: Числа уровень 3 и выше. Изучение чисел: больше, меньше, чем ... или, равны
- ✓ Геометрия 3 уровень и выше. Изучение формы и пространства: Координаты

### Навыки

- ✓ Логические рассуждения

### Возраст

- ✓ 9 лет и старше

### Материалы

Каждому ребёнку необходимо:

- ✓ Копии игровых корабликов
  - 1A, 1B для игры 1
  - 2A, 2B для игры 2
  - 3A, 3B для игры 3
- ✓ Вам также может понадобиться копии дополнительных листов игры:, 1A', 1B', 2A', 2B', 3A', 3B'.

# Кораблики

---

## Инструкция для деятельности

1. Поставьте 15 выбранных детей в одну линию перед всем классом. Дайте каждому ребёнку карточку с числом (в случайном порядке). Не показывайте числа на карточке остальным детям в классе.
2. Дайте каждому из оставшихся детей коробку с 4 или 5 конфетами. Их работа будет заключаться в поиске заданного числа. Они могут «платить» конфетами за просмотр конкретных карточек. Если они найдут правильный номер, прежде чем используют все конфеты, то получают оставшиеся.
3. Повторите, если это необходимо.
4. Затем перетасуйте карточки и раздайте их опять. На этот раз, пусть дети сортируют в порядке возрастания. Процесс поиска продолжается.
5. Если числа отсортированы, очевидная стратегия состоит в том, чтобы использовать только одну «оплату» при уменьшении в половину количества выбранных детей, при раскрытии карточки ребенка, стоящего в центре. Повторяя этот процесс, они должны быть в состоянии найти число, используя только три конфеты. Увеличение эффективности будет очевидным.

## Деятельность

Дети могут представить, в играх с корабликами, как сортирует компьютер. Так как они играют активно, то это заставляет их думать о стратегиях, которые используются при определении местонахождения судов.

# Кораблики – Игра линейного поиска

---

## Прочитайте следующие инструкции детям

1. Организуйте пары. У одного из вас есть лист 1А , у другого лист 1В . **Не показывайте** свой лист вашему партнёру!
2. Каждый из вас отметьте кружком один кораблик на верхней линии вашего игрового листа и назовите вашему партнёру его номер.
3. Потом поменяйтесь ролями и выскажите предположение, где судно вашего партнёра. (Вы говорите букву любого судна, а ваш партнер называет число соответствующее этой букве).
4. Сколько шагов требуется для определения местоположения судна вашего партнёра? Это и будет вашим счётом в игре.

(Листы 1А' и 1 В' - дополнительные листы для детей, у которых есть желание продолжить игру или кто «неосторожно» увидел лист партнёра. Листы: 2А', 2В' и 3А', 3В', необходимы для продолжения игр ).

## Проведите обсуждение

1. Какой был счёт?
2. Какой возможен минимальный и максимальный счёт? ( Они укажут на то, что - это 1, или 26, соответственно. Предполагаем, что дети не будут дважды выбирать то же самое судно. Этот метод называют «линейным поиском», поскольку перебираются все местоположения, один за другим).

# Кораблики – Игра бинарного поиска

---

## Инструкции

Инструкции для этой версии игры такие же, как и для предыдущей. Но номера судов теперь расположены в порядке возрастания. Объясните это условие детям перед началом игры.

1. Организуйте пары. У одного из вас есть лист 2А, у другого лист 2В. **Не показывайте** свой лист вашему партнёру!
2. Каждый из вас отметьте кружком один кораблик на верхней линии вашего игрового листа и назовите вашему партнёру его номер.
3. Потом поменяйтесь ролями и выскажите предположение, где судно вашего партнёра. (Вы говорите букву любого судна, а ваш партнёр называет число соответствующее этой букве).
4. Сколько шагов требуется для определения местоположения судна вашего партнёра? Это и будет вашим счётом в игре.

## Проведите обсуждение

1. Какой был счёт?
2. Какая стратегия использовалась для получения наименьшего счёта?
3. Какое судно вы должны были выбрать первым? (Один говорит, что то, которое – посередине, которое определяет границу части линии, где должно располагаться искомое судно).
4. Какая следующая стратегия выбора местоположения? (Опять лучшая стратегия состоит в том, чтобы выбрать середину линии и ту секцию, в которой расположено искомое судно).
5. Если применялась эта стратегия, то, сколько шагов потребовалось, чтобы определить судно? (Самое большее – пять). Этот метод называют «двоичным поиском», потому что это проблема решается - делением на две части.

# Кораблики – Игровой поиск с применением хеширования

---

## Инструкции

1. Каждый из вас берет лист, как в предыдущих играх, и называет вашему партнёру число выбранного им судна.
3. В этой игре вы можете определить в какой колонке (0 до 9 ) находится судно. Вы просто прибавляете все цифры номера судна. Последняя цифра суммы – колонка, в которой находится судно. Например, определяем местонахождение судна с номером 2345, складываем цифры  $2+3+4+5$ , получаем 14. Последняя цифра суммы 4, поэтому искомое судно должно находиться в 4 колонке. Как только узнаете колонку, вам надо догадаться, какое из судов в этой колонке является искомым. Эту технологию называют «хеширование», потому что цифры перемешены («хешированы») вместе.
4. Теперь играйте в игру, используя эту новую стратегию. Вы, может быть, захотите поиграть в эту игру более чем один раз, тогда используйте тот же самый лист, только выбирайте различные колонки.

(Заметьте, что существует отличие от других игр, запасные листы 3А' и 3В' должны быть использованы в паре, потому что образец судов в колонках должен переписываться).

## Проведите обсуждение

1. Расспросите всех о счете и проведите дискуссию, как в предыдущих играх.
5. Какие суда очень быстро определяются? (Те, которые единственны в колонке). Какие суда труднее определить? (Те, колонки которых содержат много других судов.)
6. Какой, из трёх поисковых процессов, самый быстрый? Почему?

Каковы преимущества каждого из трёх различных методов поиска? (Вторая стратегия поиска быстрее, чем первая. Но в первой, стратегии не требуется, чтобы суда были сортированы в определенном порядке. Третья стратегия обычно быстрее, чем предыдущие две, но возможно, в случае неудачного выбора начального судна, может быть очень медленной.

В худшем случае, если в завершении процесса все суда окажутся в той же самой колонке, то эта стратегия поиска будет такой же медленной, как и первая).

# Расширенная деятельность



























---

1. Пусть дети придумают свои игры с использованием предложенных трех форматов. Для второй игры они должны представить числа в порядке возрастания. Спросите, как они могли бы сделать очень трудной Хеш-Игру. (Самая трудная игра – та, в которой все суда расположены в одной и той же колонке). Как в этой ситуации поступить наиболее лучшим способом? ( Вы должны попытаться поместить некоторое число судов в отдельную колонку).
2. Что бы произошло, если искомого судна не было в игре? (В Игре линейного поиска - потребовалось бы - 26 шагов, чтобы показать это. В Игре бинарного поиска - необходимо было пять шагов, чтобы это доказать. При использовалась Системы Хеширования - всё зависело бы от того, сколько судов появились бы в соответствующей колонке).
3. Используя стратегию Бинарного поиска, сколько шагов потребовалось бы, чтобы определить 100 местоположений (приблизительно шесть), 1000 местоположений (приблизительно девять), или миллион (приблизительно девятнадцать)? (Заметьте, что число шагов увеличивается очень медленно, по сравнению с числом судов. Один дополнительный шаг требуется каждый раз, когда размер удваивается, таким образом, число шагов пропорционально логарифму числа судов.)



# My Ships

## Number of Shots Used:

 9058	 7169	 3214	 5891	 4917	 2767	 4715	 674	 8088	 1790	 8949	 13	 3014
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
 8311	 7621	 3542	 9264	 450	 8562	 4191	 4932	 9462	 8423	 5063	 6221	 2244
<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>

# Your Ships

## Number of Shots Used:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

My Ships				Number of Shots Used:								
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
A	B	C	D	E	F	G	H	I	J	K	L	M
2587	7187	5258	8020	1919	141	4414	3056	9118	717	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships													Number of Shots Used:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
A	B	C	D	E	F	G	H	I	J	K	L	M																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															</

1B

My Ships		Number of Shots Used:																									
163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031															
A	B	C	D	E	F	G	H	I	J	K	L	M															
5283	5704	6025	6801	7440	7542	7956	8094	8672	9137	9224	9508	9663															
N	O	P	Q	R	S	T	U	V	W	X	Y	Z															











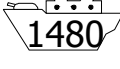

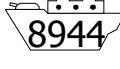
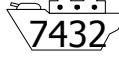

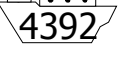
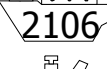









Your Ships		Number of Shots Used:																									
A	B	C	D	E	F	G	H	I	J	K	L	M															
N	O	P	Q	R	S	T	U	V	W	X	Y	Z															


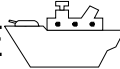
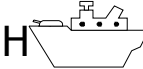
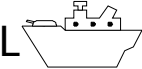

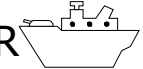
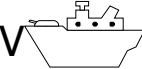
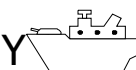


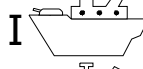
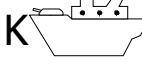




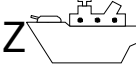


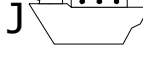
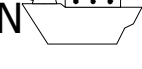
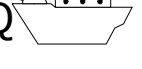

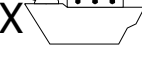
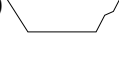

2A

My Ships													Number of Shots Used:												
33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655	A	B	C	D	E	F	G	H	I	J	K	L	M
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Your Ships													Number of Shots Used:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
A	B	C	D	E	F	G	H	I	J	K	L	M																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															</

2B

My Ships					Number of Shots Used:				
0	1	2	3	4	5	6	7	8	9
A  9047	C  3080		E  5125	H  8051	L  7116	O  6000	R  9891		W  1062
B  1829	D  9994		F  1480	I  1481	M  8944	P  7432	S  1989	V  4392	X  2106
			G  8212	J  4712	N  4128	Q  4110	T  2050		Y  5842
				K  6422		U  8199			Z  7057

Your Ships					Number of Shots Used:				
0	1	2	3	4	5	6	7	8	9
A 	E 	H 		L 		O 	R 	V 	Y 
B 	F 	I 	K 	M 		P 	S 	W 	Z 
C 	G 	J 		N 		Q 	T 	X 	
D 							U 		

3A

My Ships									
Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A 9308 B 1478 C 8417 D 9434	E 6519 F 2469 G 5105	H 1524 I 8112 J 2000	K 4135	L 9050 M 1265 N 5711		O 4200 P 7153 Q 6028	R 3121 S 9503 T 1114 U 7019	V 2385 W 5832 X 1917	Y 1990 Z 2502



























Your Ships									
Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9
A B	C D		E F G	H I J K	L M N	O P Q	R S T U	V	W X Y Z

3B

My Ships		Number of Shots Used:	
6123	1519	9024	5164
A	B	C	D
1884	3541	5251	4840
N	O	P	Q
		R	S
		T	U
		V	W
		X	Y
		Z	

Your Ships		Number of Shots Used:	
A	B	C	D
N	O	P	Q
		R	S
		T	U
		V	W
		X	Y
		Z	

1A'







My Ships													Number of Shots Used:												
 2387  9003  3951  5695  1284  4761  7118  1196  1741  3791  3405  3132  6682													A B C D E F G H I J K L M												
 9493  9864  7359  1250  7036  2916  7562  9299  8910  6713  5173  8617  4222													N O P Q R S T U V W X Y Z												

Your Ships													Number of Shots Used:												
A B C D E F G H I J K L M																									
N O P Q R S T U V W X Y Z																									

1B'



My Ships													Number of Shots Used:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
28	326	943	1321	1896	2346	2430	2929	3106	3417	4128	4717	4915																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

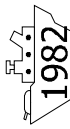



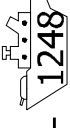
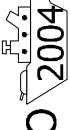
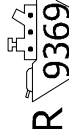





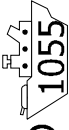





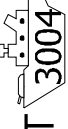






Your Ships													Number of Shots Used:												
																									
A	B	C	D	E	F	G	H	I	J	K	L	M													
																									
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

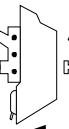
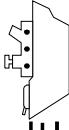
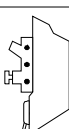
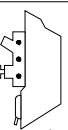
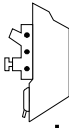
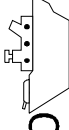
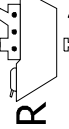
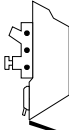
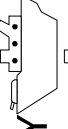
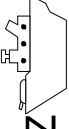
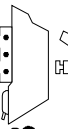
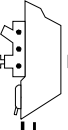

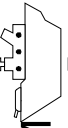
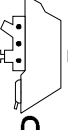
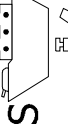
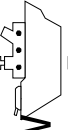

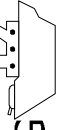

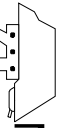
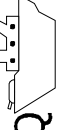

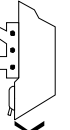
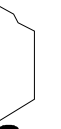
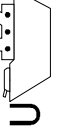
2A'

My Ships													Number of Shots Used:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

Your Ships													Number of Shots Used:												
A	B	C	D	E	F	G	H	I	J	K	L	M													
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

2B'

My Ships		Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9		
A 1982 	C 6113 		E 9121 	H 5009 	L 1248 	O 2004 	R 9369 	V 3285 	W 9172 	X 2052 	Y 6012 
B 7841 	D 1055 		F 1011 	I 2651 	M 1716 	P 5173 	S 1321 	T 3004 	U 7190 		
			G 2984 	J 1751 	N 2148 	Q 2806 			Z 7525 		

Your Ships		Number of Shots Used:									
0	1	2	3	4	5	6	7	8	9		
A 	E 	H 	K 	L 		O 	R 	V 	Y 	Z 	
B 	F 	I 		M 		P 	S 	W 			
C 	G 	J 		N 		Q 	T 	X 			
D 							U 				

3A'

My Ships		Number of Shots Used:																	
0	A 8615 B 7003 C 1991 D 6211	1	E 1361 F 7644 G 5600	2	H 7726 I 9003 J 5557	3	K 3000	4	L 1814 M 2002 N 8844	5		6	O 9656 P 4002 Q 1221	7	R 6993 S 3121 T 4300 U 1907	8	V 8208 W 9423 X 4176	9	Y 2917 Z 4122

Your Ships										Number of Shots Used:									
0		1		2		3		4		5		6		7		8		9	
A B		C D				E F G		H I J K		L M N		O P Q		R S T U		V		W X Y Z	

3B'

# О чём это всё?

---

Компьютеры хранят большой объём информации, и у них должна быть возможность её быстро просеять. Одной из самых больших мировых проблем в поиске информации, с которой сталкиваются интернет - поисковые системы, заключается в том, чтобы найти миллиарды веб-страниц в доли секунд. Данные, которые компьютер просит указать, такие как: слово, числа штрихового кода или имя автора, называют *ключом поиска*.

Компьютеры могут обрабатывать информацию очень быстро, и вы, могли бы, наверно, подумать, что для того чтобы найти информацию, они просто должны отслеживать её, с начала и до момента, пока необходимая информация не будет найдена. Это то, что мы осуществляли в Игре *линейного поиска*. Но этот метод является очень медленным, даже для компьютеров. Например, предположим, что на полках супермаркета – 10 000 различных продуктов. Когда для просмотра на контроле штрихового кода, компьютер должен просмотреть 10 000 чисел, чтобы определить наименование продукта и его цену. Даже, если это займет только тысячную долю секунды для проверки каждого кода, то, десять секунд нужно было бы для прохождения всего списка товаров. Представьте, сколько времени было бы затрачено для проверки всех товаров необходимых для семьи!

Лучшая стратегия - *двоичный поиск*. В этом методе числа сортированы в определенном порядке. Проверка средней части списка, определяет, в какой из частей расположен ключ поиска. Процесс повторяется до тех пор, пока искомый объект не будет найден. Возвращаясь к примеру с супермаркетом, эти 10 000 наименований товаров, могут быть найдены в процессе четырнадцати поисков, которые заняли бы две сотых доли секунды, т.е. практически мгновенно.

Третью стратегию нахождения данных называют - *хеширование*. Здесь ключом поиска управляют, чтобы указать точно, где найти информацию. Например, если ключ поиска – номер телефона, то вы могли бы сложить все цифры в номере телефона и взять остаток, как при делении числа 11. В этом отношении, хеш-ключ немного напоминает контрольные веса, обсуждаемые в Деятельности 4 – маленькая часть данных, значение которой зависит от других обработанных данных. Обычно компьютер мгновенно находит то, что ищет. Здесь есть маленькая вероятность, что несколько ключей окажутся в одном и том же месте, тогда компьютеру необходимо будет осуществить прямой поиск по каждому из них, до тех, пока не найдет искомый.

Программисты обычно используют для поиска стратегии хеширования, если только, нет необходимости в упорядочении данных, или возможен случайный медленный поиск.

# Деятельность 7

---

## Самый легкий и самый тяжелый - Алгоритмы сортировки

### Введение

Компьютеры часто используют списки при выполнении определенного заказа в их работе, например, необходимо расположить объекты в алфавитном порядке, или по времени указать получение электронной почты, или по другим параметрам. Сортировка списков помогает нам быстро найти и увидеть искомые объекты. Если вы сортируете списки оценок контрольной работы всего класса, становятся очевидными самые низкие и самые высокие оценки.

Если вы выбрали неправильный метод, то потребуется много времени для сортировки большого списка, даже если компьютер быстродействующий. К счастью, известны несколько методов быстрой сортировки. При выполнении заданий этой деятельности, дети познакомятся с различными методами сортировки и смогут сравнить быстроту сортировки «умных» и простых методов.

### Учебные направления

- ✓ Математика: Измерения Уровень 2 и выше. Выполнение практических задач взвешивания.

### Навыки

- ✓ Использование весов
- ✓ Сортировка
- ✓ Сравнение

### Возраст

- ✓ 8 и старше

### Материалы

Каждая группа детей должна иметь:

- ✓ Наборы из 8 контейнеров одинакового размера, но разного веса (например, коробки из-под молока или любые другие емкости, заполненные песком)
- ✓ Гири для баланса
- ✓ Лист деятельности: Сортировка контейнеров по их весу (стр. 66)
- ✓ Лист деятельности: Разделите и завоюйте (стр. 67)

# Самый лёгкий и самый тяжёлый

---

## Инструкция

Компьютерам часто приходится сортировать списки объектов. Решение задачи, определения порядка расположения объектов в списках, бывает порой неожиданной блестящей идеей. Что случится, если объекты будут неупорядочены?

Компьютеры обычно могут сравнивать одновременно только два значения. Эта деятельность дает детям представление как это происходит.

## Деятельность

1. Распределите детей по группам.
2. Каждая группа должна иметь копию листа деятельности (стр.66), свои контейнеры и гири.
3. Пусть дети выполняют задания этой деятельности и обсудят полученный результат.

# Лист деятельности: Сортировка контейнеров по их весу

**Цель:** Найти лучший способ сортировки неизвестных весовых контейнеров.

**Вам необходимо иметь:** песок или воду, 8 одинаковых контейнеров, несколько рычажных весов.

**Что делать:**

1. Наполните каждый контейнер разным количеством песка или воды. Закройте герметично.
2. Смешайте их таким образом, чтобы вы не могли распределить их по весу.
3. Определите самый легкий по весу контейнер. Какой для этого существует самый простой способ?

**Примечание:** Вам разрешено использовать весы для измерения веса каждого контейнера. Только два контейнера можно сравнивать с помощью рычажных весов одновременно.

4. Возьмите произвольно 3 контейнера и сортируйте их в порядке возрастания веса: от самого лёгкого до самого тяжелого, используя только весы. Как вы это делали? Какое минимальное число перестановок контейнеров вы сделаете? Почему?
5. Сейчас все объекты вами сортированы в порядке от самого легкого до самого тяжелого.

Когда, как вы думаете, процесс сортировки окончен - проверьте каждую из рядом расположенных пар контейнеров.

## Сортировка выбором

Одним из методов, который может использовать компьютер, называется *сортировка выбором*. Расскажем, как работает данный вид сортировки. В начале, определяется самый легкий вес и убирается в сторону. Следующий шаг: находится самый легкий вес из оставшихся весов и удаляется. Повторяется этот процесс, пока все веса не будут удалены.



Подсчитайте, сколько операций сравнения вы сделали.

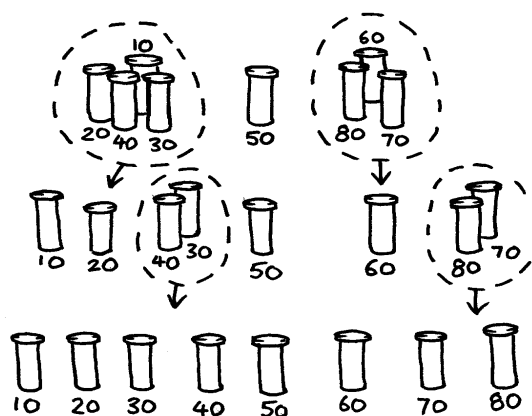
**Дополнительно для любознательных:** Вычислите математически количество операций сравнения для сортировки в определенном порядке 8 объектов? А 9 объектов? 20?



## Лист деятельности: Метод «Разделяй и властвуй» (D & C)

### Быстрая сортировка

«Quicksort» (Быстрая сортировка) намного быстрее, чем сортировка выбором, особенно при сортировке больших списков. Фактически, лучший из известных методов сортировки. Рассмотрим, как этот метод работает. Выберите наугад один из объектов и переложите его в другую сторону. Затем сравните каждый, из оставшихся объектов, с выбранным. Поместите те объекты, которые легче – слева, а более тяжелые – справа. (By chance you may end up with many more objects on one side than on the other.) (У вас может оказаться намного больше объектов на одной стороне, чем на другой, при перекладывании объектов). Выберите объекты с одной из сторон и повторите эту процедуру. Запомните тот объект, который находится в центре всей группы объектов. Продолжайте повторение этой процедуры до тех пор, пока в одной из групп останется не более одного объекта. Как только все объекты будут разделены на группы, одна из которых будет состоять из единственных объектов, тогда эти объекты и будут расположены в порядке возрастания: от самого легкого до самого тяжелого.



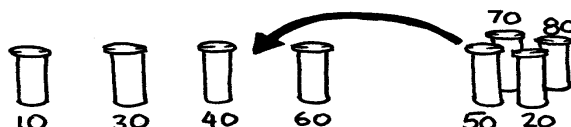
Какое количество сравнений было в этом процессе? Вы должны убедиться, что быстрая сортировка («quicksort») – будет еще более эффективна, чем метод сортировки выбором, если Вам посчастливится вначале процесса выбрать самый легкий или самый тяжелый вес. Если вы выберете удачно средний вес, то вам достаточно будет 14 сравнений, по сравнению с 28 при сортировке выбором. Во всяком случае, «quicksort» метод, никогда не будет существенно хуже, чем сортировка выбором, а при удачном старте, и может быть и на много лучше!

**Дополнительно для любознательных:** Если бы вы, допустим случайно, всегда выбирали вначале более легкий объект, то какое количество перестановок было бы?

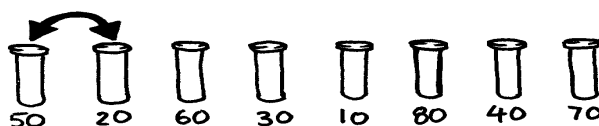
## Вариации и расширения

Было придумано много различных методов сортировки. Вы можете попытаться сортировать свои весовые контейнеры следующими методами:

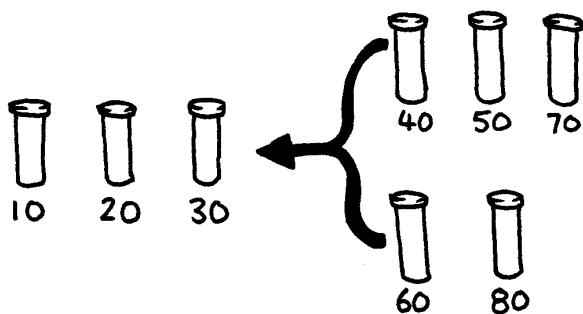
*Сортировка вставкой* осуществляется удалением каждого объекта из несортированной группы и вставка его, с коррекцией положения, в растущий список (см. картину ниже). С каждой вставкой уменьшается группа несортированных объектов, а сортированный список растет, пока, в конечном счете, весь список не будет сортирован. Карточные игроки часто используют этот метод, чтобы руками сортировать всю колоду.



*Сортировка пузырьком* состоит в повторяющихся проходах по всему списку, и сравнении попарно стоящих объектов друг с другом, если порядок в паре неверный, выполняется перестановка объектов. Проходы по списку повторяются до тех пор, пока на очередном проходе не окажется, что больше не происходит никаких перестановок. Этот метод не очень эффективен, но некоторым людям он кажется более понятным, чем другие.



*Сортировка слиянием* - еще один метод «D&C» упорядочения объектов списка. В-первых, список разделяется произвольным образом на два списка, примерно одинакового размера (или почти одинакового, если количество элементов списка четное). Каждая из получившихся частей сортируется отдельно, и уже, два упорядоченных списка соединяются в один. Сортировка слиянием двух массивов объектов происходит легче, если вы неоднократно в этих двух массивах удалите меньшие по весу объекты на начальном этапе сортировки. На рисунке внизу показано, что 10, 20 и 30 – граммовые контейнеры выставлены вначале списка, таким образом, следующим в список, будет добавлен 40-граммовый контейнер. Как вы сортируете маленькие списки? Для маленьких – только и используют данный вид сортировки! В итоге, все списки будут разбиты на отдельные элементы, т.о., вам не нужно беспокоиться о том, когда необходимо остановиться.



# О чём это всё?

---

Информацию намного легче найти в отсортированном списке. В телефонных справочниках, словарях и книжных каталогах используют алфавитную индексацию, и было бы намного труднее жить, если бы это не делали. Если список с числами (таким, как список расходов) отсортирован в определенном порядке, то экстремальные значения легче увидеть, потому что, они расположены в начале и в конце списка. Одинаковые списки также легко определить, так как они заканчиваются вместе.

Компьютеры тратят много времени на сортировку элементов, поэтому программистам необходимо найти быстрые и эффективные методы. Некоторые из более медленных методов, такие как сортировка вставкой, выбора или пузырька используются в специальных случаях, но обычно используют метод быстрой сортировки.

В методе быстрой сортировки используется понятие, названное рекурсией. Это понятие означает, что вы разбиваете список на меньшие списки, а затем осуществляете такую же сортировку в каждом из этих, вновь полученных, списков. Этот особый подход называют «разделяй и властвуй «D&C»».

Список делится до тех пор, пока он не будет достаточно мал, чтобы его «завоевать». В быстрой сортировке, списки разделяют до тех пор, пока они не будут содержать единственный элемент. Это тривиальная сортировка единственного элемента! Хотя это кажется очень сложным, но на практике этот метод сортировки быстрее остальных.

## Решение и подсказки

---

1. Лучшим способом найти наилегчайший вес – это обойти все объекты в списке по порядку, отслеживая самый легкий объект по сравнению с остальными. Таким образом, вы сравниваете два объекта и выбираете из них который легче. Повторяете до тех пор, пока все объекты не будут проверены.
2. Сравните веса, балансируя их. Это может быть легко сделано тремя сравнениями, а иногда достаточно и двух, если дети поймут, что сравнение транзитивно (что если А легче чем В, и В легче чем С, то А легче чем С).

### Любознательным:

Вот короткий путь для сложения количества сравнений в сортировке выбором.

Чтобы найти минимальный, из двух объектов, - вам надо одно сравнение; из трёх – два; из четырех - три, и так далее. Для сортировки восьми объектов необходимо сортировка выбором из 7 сравнений; чтобы найти, что первый; шесть – найти следующий; пять - найти следующее и так далее. Это выглядит так:

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ сравнений.}$$

Для  $n$  объектов  $1 + 2 + 3 + 4 + \dots + n - 1$  сравнений в сортировке.

Если мы перегруппируем числа, то сложение будет осуществляться проще.

Например, сложить числа  $1 + 2 + 3 + \dots + 20$ , сгруппируем их так:

$$\begin{aligned} &(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) + \\ &(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11) \\ &= 21 \times 10 \\ &= 210 \end{aligned}$$

Тогда, сумма  $1 + 2 + 3 + 4 + \dots + n - 1 = n(n - 1)/2$ .

# Деятельность 8

---

## Бой часов — Сортировка сетей

### Введение

Большая скорость работы компьютера, не всегда гарантирует быстрое решение некоторых проблем. Одним из способов увеличения скорости является использование нескольких компьютеров при решении разделенной на части поставленной проблемы. В деятельности 8, мы используем сортировку в сети, т.е. определенное количество перестановок в процессе сортировки за один и тот же промежуток времени.

### Учебные направления

- ✓ Математика: «Числа», 2 уровень и выше.
- ✓ Исследование чисел: больше чем, меньше чем

### Умения и навыки

- Сравнение
- Классификация
- Развитие алгоритма
- Комплексное решение проблемы

### Возраст

- ✓ От 7 лет и старше

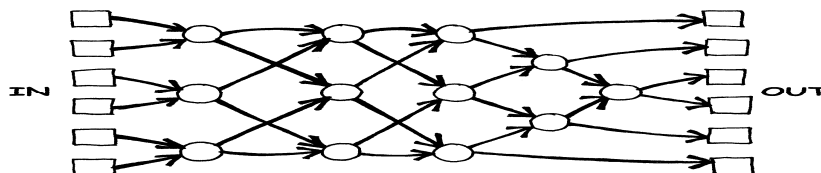
### Материалы

Для внеклассной деятельности группы (например, на пришкольном участке, в спортзале и др.)

- Мел
- Два набора из шести карт
- Фотокопия: «Сортировка сетей»(стр. 73), разрезанные карточки
- Секундомер

# Сортировка сетей

Предварительно начертите мелом на площадке сеть, изображенную на рисунке ниже на рисунке.

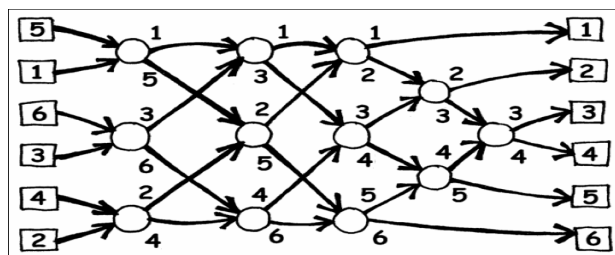


## Инструкция для детей

При выполнении этой деятельности вы увидите, как компьютер сортирует случайные числа в определенном порядке, используя принцип, называемый сортировкой сети.

1. Распределитесь по командам из шести человек. Только одна команда за один раз использует сеть.
2. Каждый член команды берет пронумерованные карточки.
3. Каждый член команды занимает свой квадрат на старте - слева на площадке, как указано на рисунке. Ваши числа должны быть расположены беспорядочно.
4. Вы движетесь в направлении, указанном стрелками до круга, где вы должны встретиться с другим членом команды.
5. Когда другой член команды приходит в тот же круг, то вы сравниваете ваши карты. Человек с карточкой, на которой написано меньшее число – продолжает движение, указанное стрелкой, - левее. Если на вашей карточке, число больше чем у вашего друга, то вы продолжаете движение по стрелке - правее.
6. Вы правильно совершаете движение, когда финишируете на другой стороне площадки?

Если команда делает ошибку, дети должны начать снова. Проверьте, что вы поняли как вам действовать в узлах (кругах) сети, а именно, если у вас меньшее значение на карточке – то влево, а если другое – то вправо. Например,



## Фотокопии для инструктора: Сортировка сетей

---

**1**

**2**

**3**

**4**

**5**

**6**

---

**156**

**221**

**289**

**314**

**422**

**499**

## Различные вариации деятельности

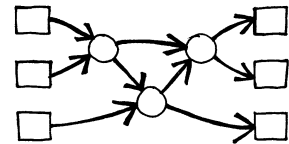
1. Если дети умеют пользоваться секундомером, предложите им использовать его для подсчета времени, затраченного каждой командой при прохождении сети.
2. Используйте карточки с большими числами (например, трехзначными числами, как предложено в фотокопии инструктора).
3. Изготавливайте карточки с очень большими числами, при сравнении которых детям надо будет приложить определенные усилия, а также используйте буквы (или слова) для сравнения в алфавитном порядке.

## Расширение деятельности

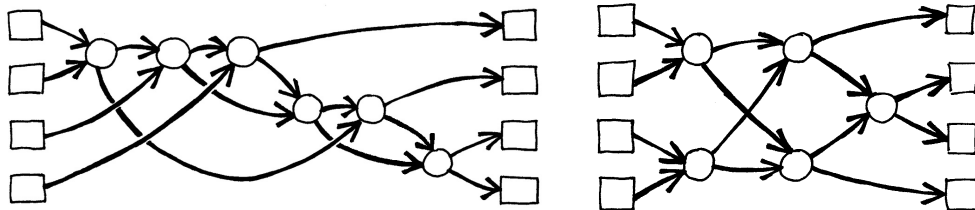
1. Что будет происходить, если дети, у которых карточки с меньшим числом, будут двигаться не влево, а вправо, (или наоборот)? (Числа будут сортированы в обратном порядке.)

Этот процесс выполняется, если сеть используется для сортировки в другом направлении? (Это не всегда будет выполняться, и дети должны привести пример стартовой позиции, которая приводит к неправильной сортировке.)

2. Попробуйте спроектировать меньшие или большие сети. Например, вот сеть, в которой сортируют три числа. Дети должны попытаться придумать примеры самостоятельно.

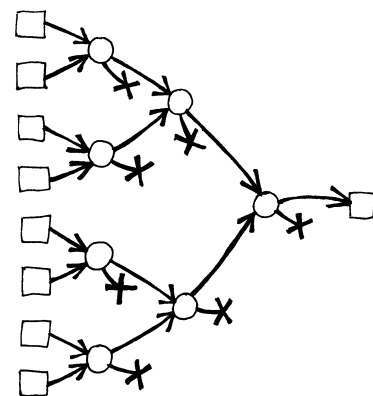


3. Ниже приведены примеры двух сетей, которые сортируют четыре входа. Какая из них сортирует быстрее? (Вторая. Принимая во внимание, что в первой требуется, чтобы все сравнения были сделаны последовательно, один за другим. Во второй, сравнения происходят одновременно. Первая сеть – пример последовательного процесса, тогда как, вторая - использует параллельный процесс, что значительно быстрее.)



4. Попробуйте осуществить сортировку большой сети.

5. Сети могут быть использованы для нахождения максимального или минимального значения входных данных. Например, вот сеть с восемью входными данными, и единственным значением на выходе, в которой будут минимальные значения на входе (остальные значения будут оставаться в тупиках/листьях сети).



6. Какие процессы, в повседневной жизни, могут или не могут быть ускорены, при использовании принципа параллелизма? Например, процесс приготовления пищи – медленный, так как на каждом этапе добавляется только один ингредиент блюда и последовательно: один за другим. Какой вид работы может быть выполнен быстрее привлечением большего количества работников? А какой нет?



# О чём это всё?

---

Так как мы используем компьютеры все больше и больше, то хотим, чтобы они обрабатывали информацию как можно быстрее.

Один из способов ускорения этого процесс - является написание компьютерной программы, которая использует меньшее количество вычислительных шагов (как показано в Деятельности 6 и 7).

Другим способом быстро решить проблему - это несколько работающих компьютеров, которые находят решения отдельных частей одной и той же проблемы за один и тот же промежуток времени. Например, производится сортировка в сети шести чисел, несмотря на то, что в итоге использовались 12 сравнений при сортировке этих чисел, три сравнения выполнены одновременно. Это значит, что затраченное время на данную сортировку будет такое же, как и для сортировки с 5 шагами сравнения. Эта параллельная сеть сортирует списки в два раза быстрее, чем любая другая сетевая система, выполняющая только одно сравнение за тот же промежуток времени.

Но не все задачи могут быть решены быстрее с использованием параллельного вычисления. Так в аналогии, представьте одного человека, который роет котлован длиной десять метров. Если бы ему помогали десять человек, и каждый бы вырыл бы по одному метру, то работа была бы завершена намного быстрее. Однако, та же стратегия не применима для рытья вглубь, например, колодца, так как следующий метр не доступен, пока не вырыт предыдущий. Программисты всё же активно пытаются найти лучшие способы разбиения проблемы так, чтобы они были решаемы компьютерами, работающими параллельно.

# Деятельность 9

---

## Грязный город – *Минимальный обход деревьев*

### Введение

Общество связано огромным количеством сетей: телефонные сети, сервисные системы поставок, компьютерные сети, а также дорожные сети. В практике сетей важен выбор того, как и куда дороги, кабели или линии связи будут проложены. Нам необходимо найти эффективные способы соединения объектов в сети.

### Учебные направления

- ✓ Математика: Геометрия от 2/3 уровня и старше. Исследование формы и пространства: обнаружение кратчайших путей на различных картах.

### Возраст

- ✓ От 9 лет и старше

### Умения и навыки

- ✓ Решение проблемы

### Материалы

Каждому ребенку необходимо:

- ✓ Лист деятельности: Проблема грязного города (стр. 78)
- ✓ Рабочая поверхность или квадраты из картона (приблизительно 40 на ребенка)

# Грязный город

---

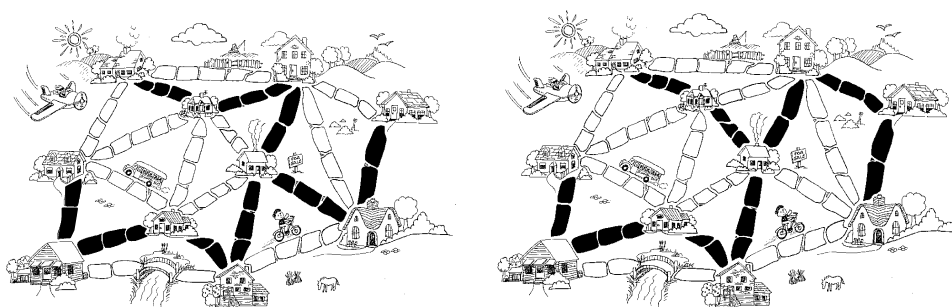
## Инструкция

В процессе этой деятельности, дети должны понять, как используются компьютеры для решения практических проблем, например, таких как: как связать здания линиями электропередач. Сделайте для детей лист деятельности «Грязный город», как показано на странице 78.

## Последующее обсуждение

Примите участие в обсуждении решений проблемы, найденных детьми. Какие стратегии ими были использованы?

Одна из стратегий нахождения лучшего решения состоит в том, чтобы начать с пустой карты и постепенно прокладывать пути, пока все здания не будут связаны друг с другом. Причем, увеличивая длину линии, не связывать этой линией те здания на рисунке, которые уже связаны. Различными решениями одной и той же проблемы: при прохождении от начала до конца, будут те, в которых изменение движения происходит путем добавления других линий пути, той же самой длины, как показано на рисунке ниже.



Другая стратегия состоит в том, чтобы начать движение во все возможные направления, а затем удалить те линии движения, которые вам не нужны. В этой стратегии вам необходимо будет приложить больше усилий.

Где в реальной жизни вам довелось иметь дело с сетями?

ИТ-специалисты называют «графами» представления этих сетей. Реальные сети могут быть представлены графом при решении таких проблем, как проектирование лучшей сети дорог между двумя населенными пунктами, или полетов самолета по всей стране.

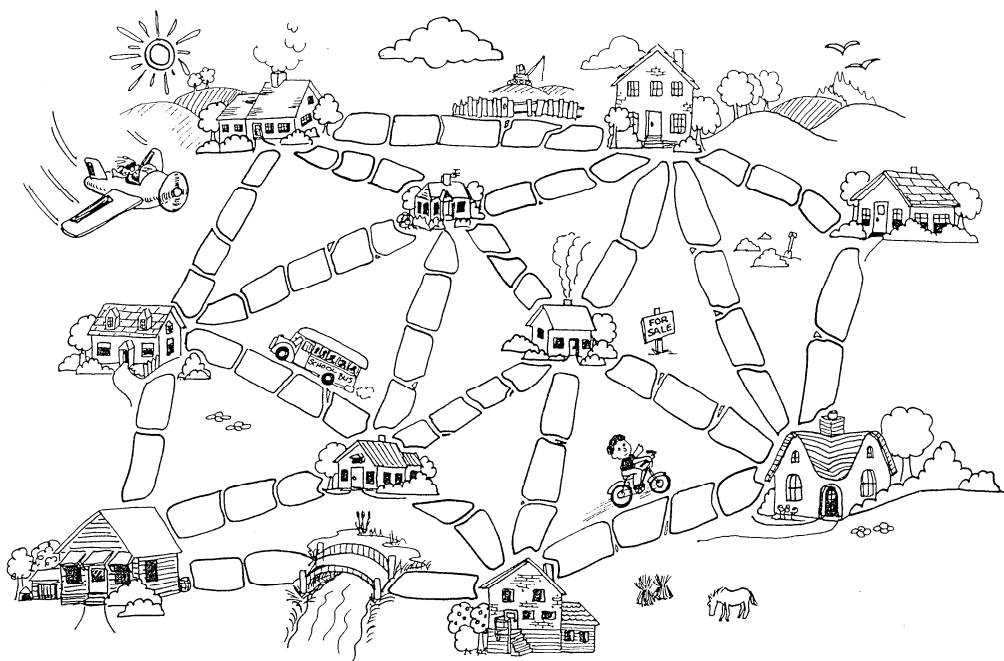
Имеются и другие алгоритмы, которые могут быть применены к графам, такие как, например, определение самого короткого расстояния между двумя пунктами, или самого короткого маршрута, который возможен при обходе всех пунктов.

## Лист деятельности: Проблемы грязного города

Когда-то давно был город, в котором не было никаких дорог. Хождение по городу было особенно трудным после ливней, потому что земля становилась очень грязной — автомобили застревали, и люди ходили в грязных ботинках. Мэр города решил, что необходимо вымостить дороги, но не хотел тратить больше денег, чем необходимо, так как городу нужен был еще и бассейн. Поэтому Мэр поставил два условия:

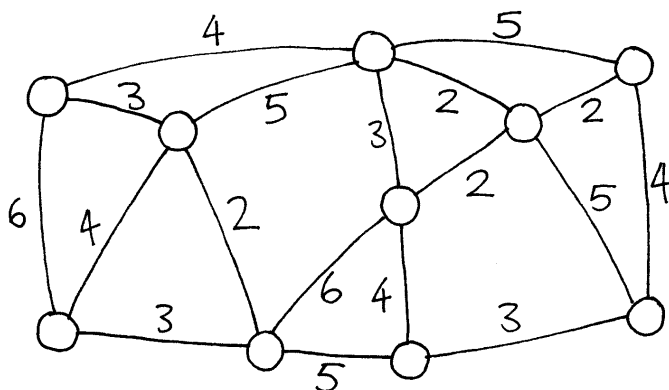
1. Достаточное количество вымощенных дорог так, чтобы каждый мог бы проехать от своего дома до любого другого, по вымощенной дороге.
2. Вымощенных дорог должно быть как можно меньше.

Здесь расположен город. Количество булыжников между каждым домом равно стоимости работы по прокладке дороги по выбранному маршруту. Определите оптимальный маршрут, который позволит соединить все здания, но при минимальных затратах (минимум количества булыжников), насколько это возможно. Какую стратегию вы использовали при решении этой проблемы?



## Вариации и расширения

Вот другой способ представления городов и дорог:



Здания представлены кругами, грязные дороги – линиями, а числа на линии указывают длину участка дороги.

ИТ-специалисты и математики часто используют такой вид диаграммы, чтобы представить подобные проблемы. Они называют это – *графом*. Это может вначале вас запутать в понимании «графа», так как «граф» иногда используется в статистике для показа различных данных в виде диаграммы или гистограммы, но программисты и математики по-другому понимают «граф». При построении линий графа масштаб не должен учитываться.

Нарисуйте свой грязный город и предложите решить его проблемы своим друзьям.

Можно сформулировать правило, которое определяет: сколько дорог или связей необходимо для лучшего решения проблемы? Это зависит от количества зданий в городе?

# О чём это всё?

---

Предположите, что вы проектируете систему, таких полезных вещей в современном обществе, как электричества, газа, или воды. Сеть проводов или труб необходима для соединения всех зданий с коммунальным предприятием. Каждый дом необходимо связать сетью, при этом не имеет значения расстояние между зданиями, так как маршрут уже существует.

Задачу проектирования сети с минимальной суммарной длиной пути называют проблемой *минимального обхода деревьев*.

Минимальный обход деревьев не только успешно применяется в городских сетях, но также помогает нам решить проблемы компьютерных сетей, телефонных сетей, нефтепроводов и маршрутов авиалиний. Однако, предлагая оптимальные маршруты для путешествующих людей, вам необходимо рассчитать не только стоимость этого маршрута. Никто не хочет провести много часов в самолете, маршрут которого к новой стране занимает большой промежуток времени, даже если билеты на него дешевле. Алгоритм «Грязный город», возможно, в этом случае, не применим.

Существуют эффективные алгоритмы (методы) для решения проблемы минимального обхода деревьев. Простой метод, который дает оптимальное решение, должен начинаться без учета связей, и только в при добавлении других частей сети, по мере её увеличения, устанавливаются дополнительные связи. Это алгоритм называют алгоритмом Дж. Краскала (Kruskal's), после его публикации J.B. Kruskal в 1956.

При решении многих проблем на графах, включая «проблему продавца путешествий», программисты и математики находят лучшие методы и быстрые алгоритмы.

## Решения и подсказки

---

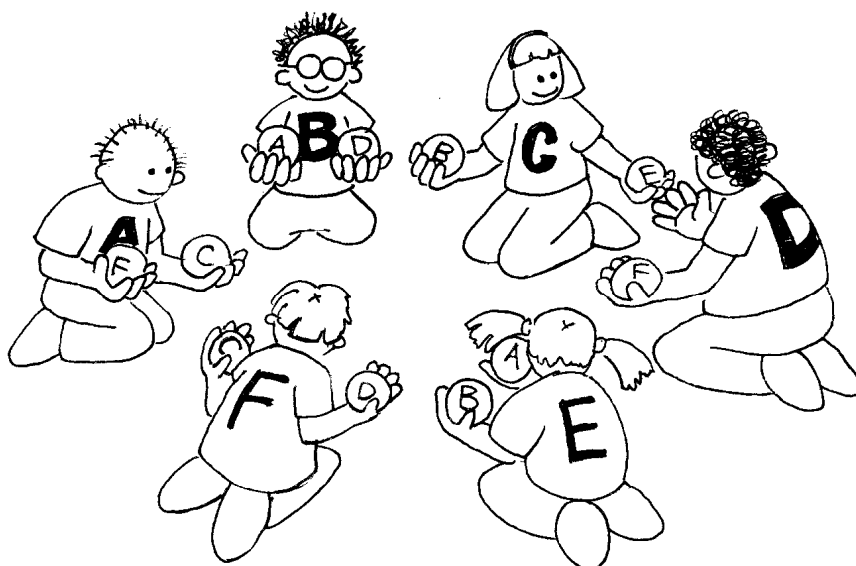
### Вариации и расширения (стр. 79)

Сколько необходимо связей (дорог) при наличии  $n$  зданий в городе? Оказывается, для оптимального решения, будет достаточно  $(n-1)$ -на связь для соединения  $n$  зданий, и добавление еще одной связи создало бы ненужные альтернативные маршруты.

# Деятельность 10

---

## Апельсиновая игра—Маршрутизация и Тупик в Сетях



### Введение

Если у вас несколько человек используют один и тот же ресурс (как например, автомобили, использующие дороги или сообщения приходящие посредством Интернета), то появляется вероятность «тупика». Совместная работа позволяет избежать тупикового случая.

### Учебные направления

- ✓ Математика: Развитие логического мышления

### Умения и навыки

- ✓ Коллективное решение проблемы
- ✓ Логические рассуждения

### Возраст

- ✓ От 9 лет и старше

### Материалы

Каждому ребёнку необходимо:

- ✓ Два апельсина или теннисных мяча
- ✓ Бейджик или этикетка

# Апельсиновая игра

## Инструкция

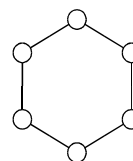
В этой игре проблема решается коллективно. Цель игры для каждого участника: получить апельсин с той же буквой, что и на его бейджике.

1. Группа из пяти и более детей садится в круг.
2. Дети прикрепляют к себе буквы алфавита (с использованием бейджика или этикетки). Каждый из детей имеет по два промаркированных (как показано на рисунке, буквами) апельсина, за исключением одного из детей, у которого только один апельсин, а другая рука свободная.
3. Распределите случайным образом апельсины между детьми. У каждого из детей в руках по два апельсина, кроме одного ребенка, у которого только один. (Ни у одного из детей не должно быть апельсина с той же буквой, что и, на их бейджиках).
4. Дети передают друг другу апельсины до тех пор, пока у них в руках не окажется апельсин с той же буквой, что и на их бейджике. Вв должны руководствоваться двумя правилами:
  - а) Только один апельсин может быть передан в одни руки.
  - б) Один апельсин можно передать по кругу в руку соседа. (Каждый ребенок может передать соседу любой из своих двух апельсинов).

Дети быстро сообразят, что если они будут «жадными» (т.е. будут стараться удерживать полученные ими апельсины), то вся группа не сможет достигнуть поставленной цели. А может быть детям намекнуть, что индивидуалисты не «выигрывают» в игре, но что приз получит каждый, если у него в руках будет апельсин с той же буквой, что и на его бейджике.

## Организация обсуждения

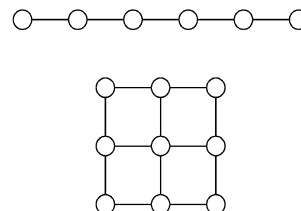
Какие стратегии дети использовали при решении проблемы? Где в реальной жизни Вы могли встретиться с ситуацией тупика? (Некоторыми примерами, могли бы быть: дорожные пробки, действия игроков вокруг мяча в бейсболе, или попытка пройти одновременно, через дверной проем большому количеству людей.)



## Расширение деятельности

Попробуйте данный вид деятельности с меньшим количеством участников или большим количеством.

- Придумайте для детей новые правила игры.
- Выполните деятельность без любой подсказки.
- Попробуйте различные конфигурации при расположении детей: в линию; при наличии у некоторых детей более двух соседей (как показано на рисунке).





# О чём это всё?

---

Маршрутизация и тупик - проблемы многих сетей, таких как дорожные системы, телефонные и компьютерные системы. Инженеры тратят много времени на проектирование сетей, в которых данные проблемы легче решаются.

Маршрутизация, поток и тупик могут представить трудноразрешимые проблемы для разного вида сетей. Только подумайте о своей поездке в час-пик! Это происходило несколько раз в Нью-Йорке, когда движение по улицам, настолько увеличивалось, что образовывался тупик: никто не мог проехать на своем автомобиле! Иногда, когда компьютеры в фирмах, работают с меньшей скоростью, чем обычно (например, в банках), то говорят, что проблема вызвана тупиком коммуникационной сети. Проектирование сети так, чтобы маршрутизация проходила легко и эффективно и потоки были минимизированы - трудная проблема, над которой работают инженеры различной специализации.

Иногда более, чем один человек, имеет потребность в получении одних и тех же данных за один и тот же промежуток времени. Если часть данных (таких как, банковское сальдо клиента) обновляется, важно "заблокировать" процесс в течении, когда происходит обновление. Если не заблокировать, то кто-то может производить обновления в то же время, и баланс может быть указан неверно. Однако, если это блокировка произойдет одновременно с другой блокировкой, то может быть тупик.

Одним из самых захватывающих событий в компьютерном мире - появление параллельного вычисления, где сотни или тысячи РС подобных процессоров объединены (в сети), чтобы образовать один мощный компьютер. Много проблем, наподобие, «Оранжевой Игры», должны решаться в сетях (намного быстрее!) этих непрерывно работающих параллельно компьютеров.

# **Часть III**

**Укажем Компьютерам что  
делать-**

***Представление Процедур***

# Укажем Компьютерам что делать

---

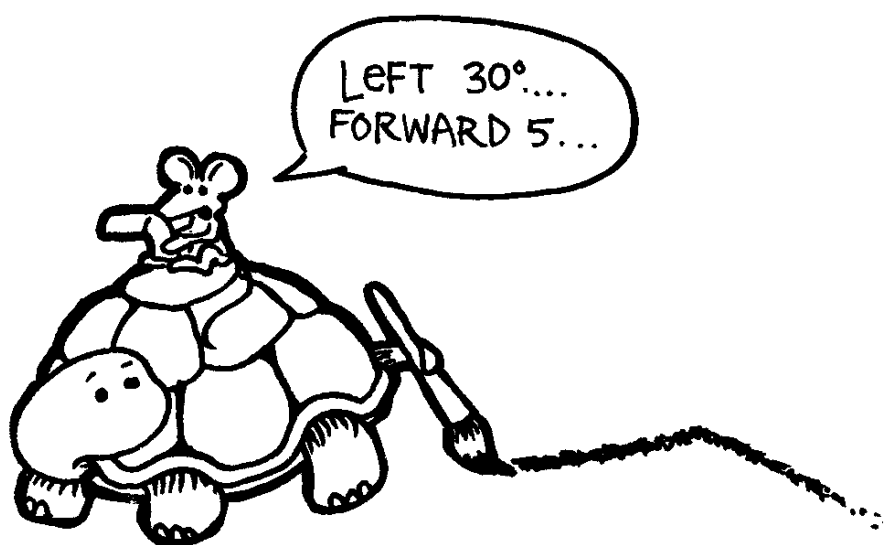
Компьютеры следуют инструкциям — миллионы инструкций каждую секунду. Сказать компьютеру, что сделать — это значит дать ему правильные инструкции, как это сделать. Но это совсем не так легко, как это звучит!

Когда мы получаем инструкции, то интерпретируем их, в соответствии со здравым смыслом. Если кто-то говорит, "проходите через эту дверь," то они не имеют в виду, что надо фактически сломать дверь — они имеют в виду, что надо пройти через дверной проем, и в случае необходимости, сначала, - открыть эту дверь! Компьютеры различные. Действительно, когда они присоединены к мобильным роботам, то необходимо предпринимать всевозможные меры безопасности, чтобы исключить возможность причинения ими убытков и вреда при буквальном интерпретировании инструкций, например, как пройти через двери. Контакт с чем-то, что повинует только инструкциям, без "обдумывания", вырабатывает определенные навыки поведения.

В деятельности 11 и 12, главы 3, у вас будет возможность представить, как осуществляется контакт с механизмами, нрав которых, четко определен набором инструкций.

Во первых, нам необходимо обучиться у «механизмов», как компьютеры распознают слова, числа или последовательности символов, с которыми им приходится работать. Эти "механизмы" называют конечными автоматами.

Вторая деятельность учит нас общению с компьютером. Хороший программист должен научиться разговаривать с компьютером, используя фиксированный набор инструкций, который интерпретируется буквально. Список инструкций - это программа. Существуют разные языки программирования, на которых программист может написать эти инструкции, если у него будет желание. Но мы будем, в процессе этой деятельности, использовать простой язык, и без компьютера.



# Деятельность 11

---

## Охота за сокровищами—*Конечный Автомат*

### Введение

Компьютерным программам часто надо обработать последовательность символов, таких как буквы или слова в документе, или даже тексте другой компьютерной программы. Программисты часто для этого используют конечный автомат. Конечный автомат (КА) формирует последовательность инструкций, позволяющий компьютеру распознавать слово или последовательность символов. Мы будем работать с неким подобием конечного автомата (КА) — картами сокровищ!

### Учебные направления

- ✓ Математика: Развитие логического мышления и умения рассуждать для описания и продолжения некоторой модели, используя слова и символы.
- ✓ общественные науки
- ✓ Английский язык

### Умения и навыки

- ✓ Чтение простых карт
- ✓ Распознавание образов (моделей)
- ✓ Логика
- ✓ Последовательное выполнение инструкций

### Возраст

- ✓ От 9 и старше

### Материалы

Вам необходимо :

- ✓ Один набор островных карт (инструкции для тех, кто будет рисовать карту, необходимо скрыть!)  
Фотокопия для Учителя: Карты островов (начиная с стр. 92 и далее) разрезанные на части. Разрезать карты необходимо вдоль пунктира и склеить так, чтобы видно было название острова, а на другой стороне – инструкции.

Для каждого ребенка необходимо:

- ✓ Лист деятельности: Найдите путь к богатству на «Острове Сокровищ» (стр. 91)
- ✓ Карандаш или ручка

Для каждого ребенка для расширения этой деятельности, необходимо:

- ✓ Лист деятельности: «Остров сокровищ» (стр. 97)
- ✓ Лист деятельности: Таинственная игра с монетой (стр. 98)

# Остров Сокровищ

## Введение

Ваша цель – найти Остров Сокровищ. Дружественные пиратские корабли, проплывая по определенным маршрутам между островами в данной части мира, предлагают путешественникам туристические экскурсии. Около каждого острова находятся два корабля для экскурсий: А и В, которые предстоит вам выбрать. Необходимо найти оптимальный маршрут к Острову Сокровищ. Вы можете, на каждом из островов, около которых будете проплывать, попросить только один из кораблей: или А или В (но не два, одновременно). Матрос с острова, будет подсказывать, какой следующий корабль вы можете арендовать. Но у пиратов, ваших дружественных попутчиков, нет карты всех островов. Поэтому используйте свою карту, чтобы проложить маршрут и выбрать корабль для путешествия.

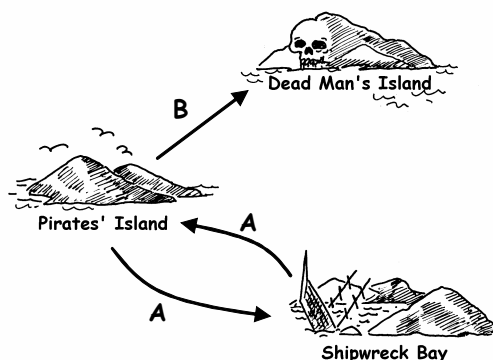
## Демонстрация

(Примечание: ниже представлены виды различных островов). Используя различные способы (проекции или доску) нарисуйте диаграмму трех островов, как здесь показано:



Скопируйте эти три карточки для каждого ребенка. (Примечание: маршруты на этих картах отличаются от маршрутов на основной карте в этом виде). Стартуйте к «Острову Пиратов» (Pirates' Island), попросив корабль «А». Каждый ребенок должен указать маршрут, к «Бухте кораблекрушений» (Shipwreck Bay). Отметьте маршрут на вашей карте. В «Бухте кораблекрушений», просите опять корабль «А». Вы будете направлены назад, к «Острову Пиратов». Отметьте это на карте. Далее, попросите корабль «В». Отметьте это на карте. Вы направляетесь к «Острову Мертвеца» (Dead Man's Island), где вам предстоит бросить якорь!

В итоге, ваша карта должна быть такой, как на этом рисунке:



## Карточки для демонстрации



Pirates' Island

A →



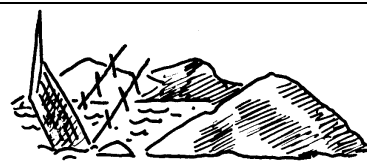
Shipwreck Bay

B →



Dead Man's Island

Pirates' Island



Shipwreck Bay

A →



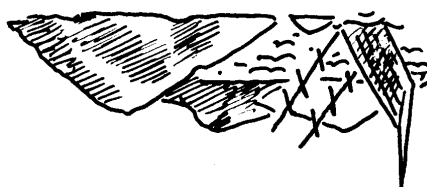
Pirates' Island

B →



Dead Man's Island

Shipwreck Bay



## Карточки для демонстрации

---



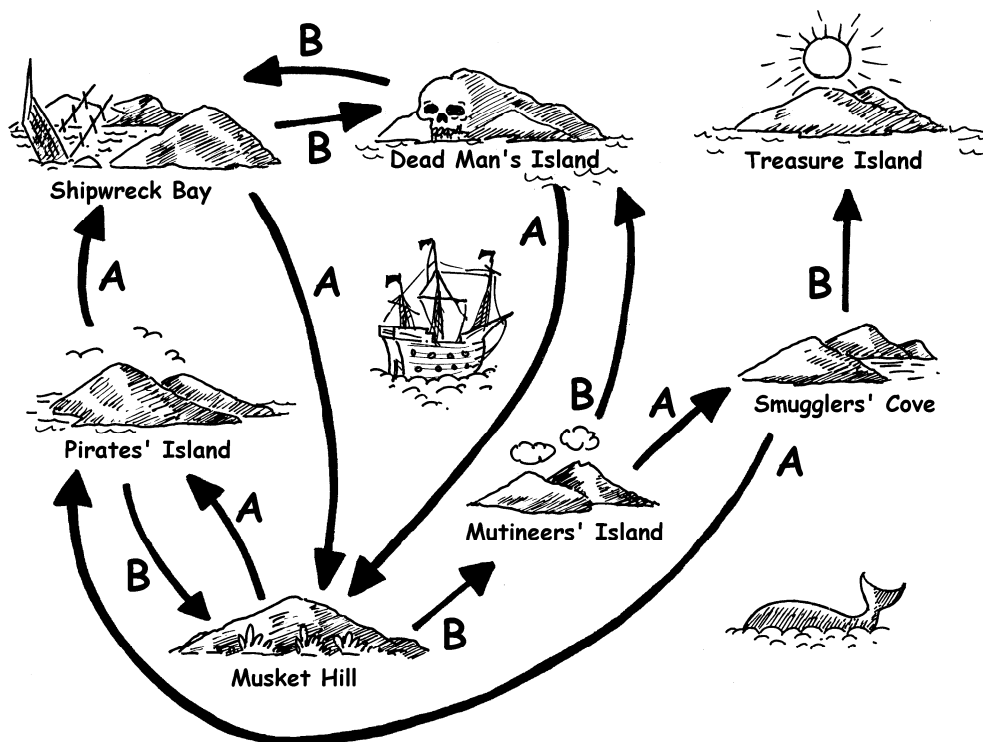
**Нет кораблей,  
отплывающих от  
Острова Мертвеца!**



## Деятельность

Выберите 7 детей, которые будут "островами". Пусть эти дети держат рисунки тех островов, которые они представляют, а на обратной стороне рисунков - секретные инструкции. Расположите детей произвольным образом в помещении или на детской площадке. Остальные ребята получают бланки карт, для нахождения оптимального маршрута от «Острова Пиратов» к «Острову Сокровищ», и отмечают его на карте. (Это - хорошая идея расставить детей по одному, чтобы они не смогли услышать о маршруте до его начала.).

Попытайтесь найти более одного маршрута. Ниже приведен пример такой карты:

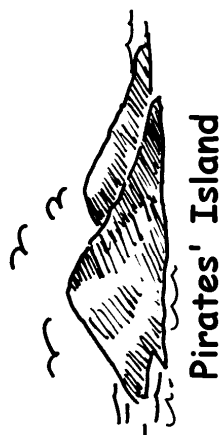
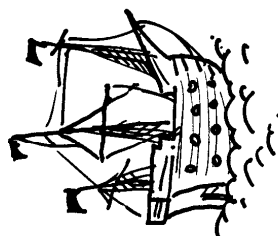
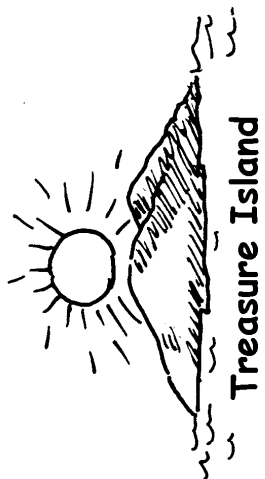


## Обсудите

Какой маршрут самый быстрый? А какой мог бы быть самым медленным? Некоторые маршруты могут иметь петли. Вы можете привести пример такого маршрута? (Например, BBVABAB и BBVABVABAB, но эти два маршрута, также приведут вас к «Острову Сокровищ».)



**Лист активности: Найдите путь к богатству на  
«Острове Сокровищ»**



# Фотокопия для учителя: Карточки островов (1/4)



Pirates' Island

A →



Shipwreck Bay

B →



Musket Hill

Pirates' Island



Shipwreck Bay

A →



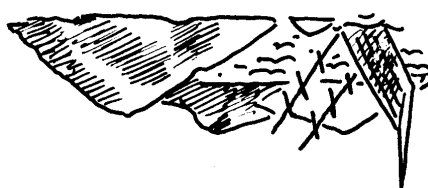
Musket Hill

B →



Dead Man's Island

Shipwreck Bay



## Фотокопия для учителя: Карточки островов (2/4)



Musket Hill

A →



Pirates' Island

B →



Mutineers' Island



Musket Hill



Dead Man's Island

A →



Musket Hill

B →



Shipwreck Bay



Dead Man's Island

# Фотокопия для учителя: Карточки островов (3/4)



Mutineers' Island

A →



Smugglers' Cove

B →



Dead Man's Island

Mutineers' Island



Smugglers' Cove

A →



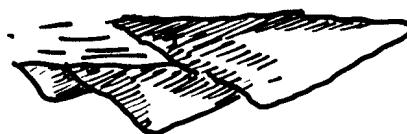
Pirates' Island

B →

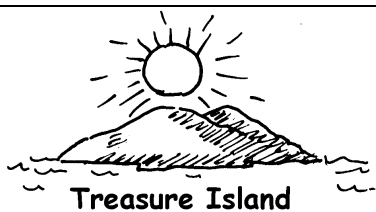


Treasure Island

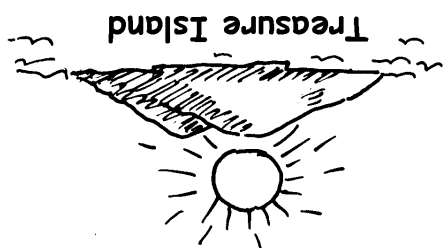
Smugglers' Cove



**Фотокопия для учителя: Карточки островов  
(4/4)**

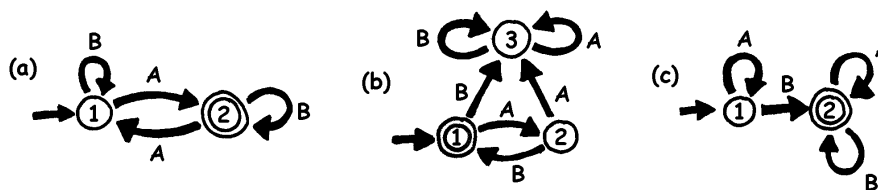


**Поздравляем!**



## Конечный автомат (КА)

Можно другим способом изобразить карту, вот так:



Острова изображены пронумерованными кругами, а остров на финише (с сокровищем) - двойным кругом. По каким маршрутам мы можем путешествовать, чтобы попасть в конечный пункт – «Остров Сокровищ»?

Примечание. По карте (a) процесс закончится немедленно (остров 2), только если в последовательности будет нечетное число кораблей А (например, АВ, ВАВАА, или АААВАВА).

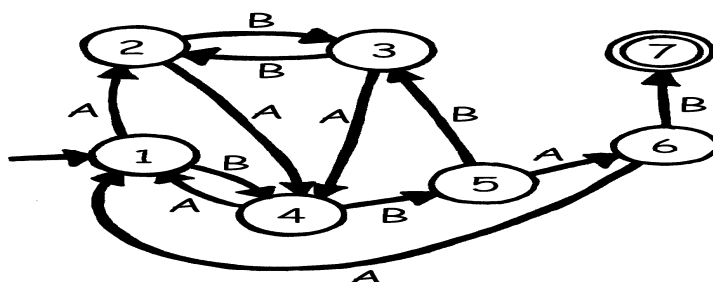
По карте (b), возможно добраться до двойного круга, только с переменной последовательностью кораблей А и В (АВ, АВАВ, АВАВАВ...).

По карте (c) необходимо, чтобы последовательность кораблей содержала, по крайней мере, один В (единственные, не подходящие последовательности, – это А, АА, ААА, АААА...).

## Лист деятельности: «Остров Сокровищ»

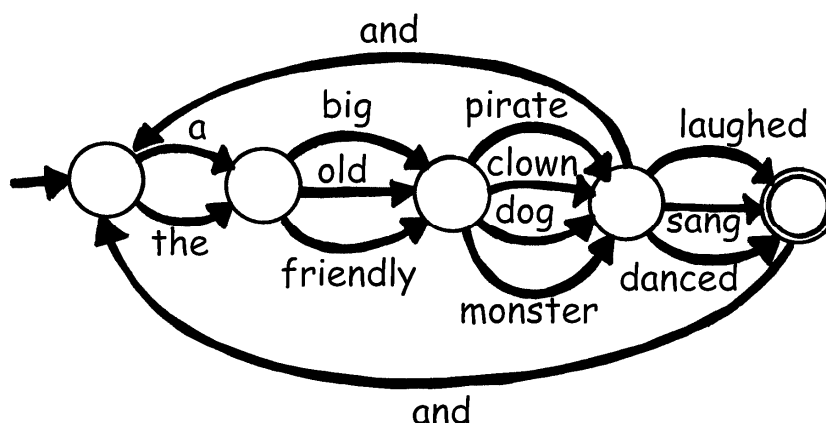
Вы можете скрыть хорошо свое захороненное сокровище? Как трудно вам найти сокровище? Пора сделать вашу собственную карту!

1. Вот более сложная версия той же самой идеи изображения карты. Эта та же карта, что и в предыдущих случаях. Программисты и математики используют этот быстрый и легкий способ проектирования маршрутов различных объектов.



Нарисуйте главный план так, что бы вы могли ясно видеть маршруты Пиратских кораблей, и составьте бланк карты и карту с изображением островов. Какая самая эффективная последовательность маршрута достижения вашего «Острова Сокровищ»?

2. Ваши друзья смогут ориентироваться по вашей карте? Предложите им последовательность A и B объектов и убедитесь, что они достигнут нужного острова.
3. Вы можете придумать много игр и загадок, основанных на идее конечных автоматов.
4. На рисунке предложен способ построения предложения, основанный на выборе слов, как показано на карте, в соответствии со случайным маршрутом.



маршрутом.

Попробуйте самостоятельно реализовать эту идею. Возможно, вы сможете составить забавную историю!

## Лист деятельности: Таинственная игра с монетой

Наши друзья загружали игру с Интернета, в которой робот подбрасывал монету, а они должны были угадать, что будет: орел или решка. Вначале игра выглядела очень легкой. По крайней мере, у них был бы 50/50 шанс на победу, или они на это надеялись! Через некоторое время, игра вызвала подозрение. Казалось, что в бросках монеты существует закономерность. Игра подстроена? Конечно, нет! Но они решили расследовать. Джо записал результаты бросков монеты, и это - то, что у них получилось: (h = орел, t = решка)

h h t h h t h h h t t h h h h t t h t t t h h h h h t h h h  
t t t h h h t t t h h h h h h t t h t t t t t h t t h t t t  
h h h t t h h h t h h h h h h h h h t t h h h t t t t h h h  
h h t t t t t t t

Вы можете найти закономерность?

Существует очень простая «карта», описывающая последовательность бросков монеты, если вы сможете увидеть на этой карте острова и маршруты кораблей. (**Подсказка:** на карте всего 4 «острова»).



# О чём это всё?

---

Конечные автоматы используются в компьютерной науке в работе с последовательностями символов или событий.

Простой пример, когда вы набираете номер телефона, то иногда получаете сообщение "Нажмите 1 для этого ... Нажмите 2 для этого ... Нажмите 3, для разговора с тем или иным человеком-оператором." Ваши нажатия клавиш есть вход в конечный автомат на другой стороне телефона. Диалог может быть очень простым или сложным. Иногда вы получаете сообщения, повторяющиеся как по кругу, так как что существуют своеобразные петли с конечным числом состояний автомата. Если это случается, то значит, что это ошибки в проектировании системы, и, вероятно, может очень расстроить звонящего!

Другим примером может быть случай с получением наличных в банкомате. Компьютерная программа ведет вас через последовательность событий. Внутри программы все возможные последовательности хранятся с конечным числом состояний автомата. Каждое нажатие ключа переводит автомат в другое положение. Некоторые из положений определяют инструкции для компьютера, например, этого "выдать 100 долларов" или "запрос состояния счета" или "извлечь банковскую карточку".

Некоторые компьютерные программы действительно составляют английские предложения с использованием карт, как рассмотрено на стр. 97. Они могут, как сами составлять предложения, так и организовать сам процесс, при вводе слов пользователями. В 1960-е годы программист написал знаменитую программу "Элиза" (по аналогии с Элизой Дулиттл), которая разговаривала с людьми. Программа выполняла роль психотерапевта, и задавала вопросы, типа "Расскажите мне о вашей семье" и "Продолжайте." Несмотря на то, что было не все "понятно", тем не менее достаточно правдоподобно, и пользователи были легковверными и думали, что они действительно разговаривали с психотерапевтом.

Хотя компьютеры, на самом деле, не очень понимают естественные языки, но они могут организовать процесс создания искусственных языков. Одним из важных типов искусственного языка является язык программирования. Компьютеры используют конечный автомат для чтения и перевод программы в элементарные инструкции, который затем могут быть "выполнены" непосредственно компьютером.

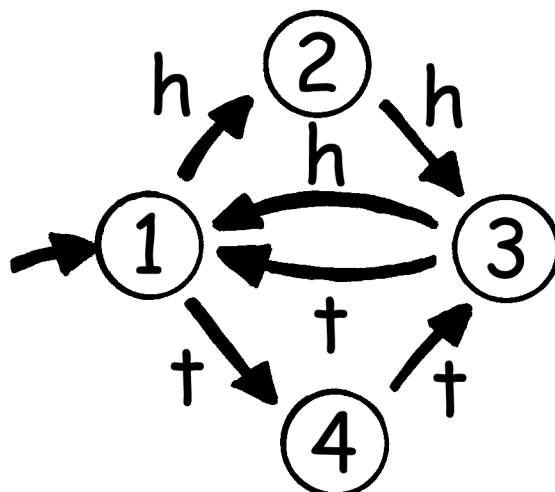


# Решения и подсказки

---

## Таинственная игра с монетой (стр. 98)

В таинственной игре с монетой используется следующая карта для бросания монеты:



Если вы будете использовать эту карту, то обнаружите, что первые два броска монеты из каждой трех, имеют одинаковый результат.

# Деятельность 12

---

## Команда вперед—Языки Программирования

### Введение

Компьютеры, как правило, используют "язык" для ввода команд, с ограниченным словарным запасом инструкций. Одним из самых неприятных особенностей программирования является то, что компьютеры всегда выполняют определенные письменные инструкции, даже если эти инструкции приводят, в итоге, к умопомрачительным результатам. В этом виде деятельности дети приобретут некоторый опыт программирования.

### Учебные направления

- ✓ Английский язык: Межличностное общение 3 уровня

### Умения и навыки

- ✓ Процедуры and инструкции.

### Возраст

- ✓ От 7 лет и старше

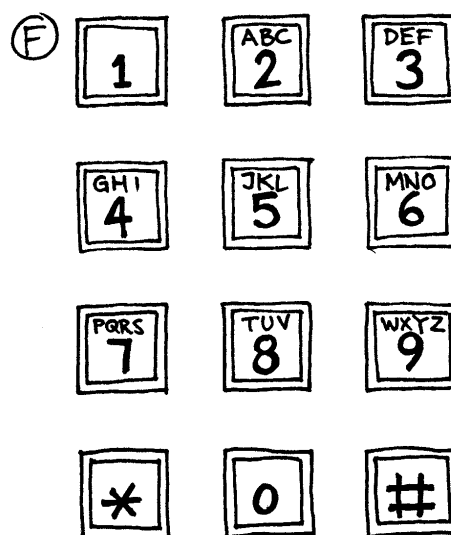
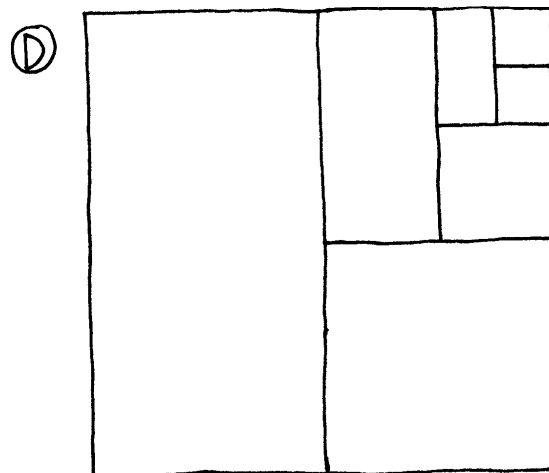
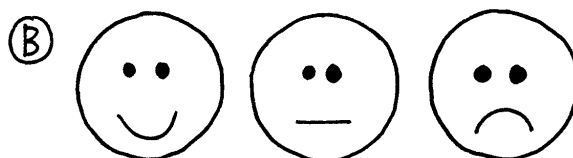
### Материалы

Вам необходимо:

- ✓ Карты с картинками, как показано на следующей странице.

Каждому ребенку необходимо:

- ✓ Карандаш, бумага и линейка



# Команда вперед

---

## Инструкция

Обсудите, всегда ли хорошо было бы, если бы люди точно выполняли инструкции. Например, что случится, если бы вы указали на закрытую дверь и сказали: "Идите через эту дверь"?

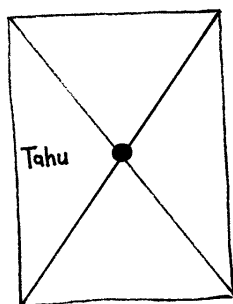
Компьютеры работают в соответствии с письменными инструкциями, даже если они не имеют смысла!

## Демонстрация примера

Посмотрите, какую картинку дети могут нарисовать, следуя инструкциям.

1. Нарисуйте точку в центре листа.
2. Начиная с верхнего левого угла страницы, проведите прямую линию через эту точку в нижний правый угол.
3. Начиная с левого нижнего угла, проведите прямую линию через эту точку в верхний правый угол.
4. Write your name in the triangle in the centre of the left-hand side of the page. Напишите свое имя в треугольник слева.

Результат должен выглядеть, примерно так:



## **Деятельность**

Выберите одного из детей, и дайте ему рисунок (например, как на стр. 102). Ребенок описывает рисунок, а класс его воспроизводит на своих листках. Дети могут задавать уточняющие вопросы. Цель этой деятельности - увидеть, насколько быстро и точно дети выполнят упражнение.

Повторите упражнение, но на этот раз дети не имеют права задавать вопросы. Лучше всего использовать для этого упражнения более простой рисунок, так как дети могут очень быстро растеряться.

Далее, попробуйте упражнение, в котором инструктируемый ребенок будет находиться за ширмой, и дети не могут ему задавать вопросы, т.е. связь между ними происходит посредством только одних инструкций.

Заметьте, что эта форма общения больше всего напоминает работу программиста при написании программ. Он задает наборы инструкций для компьютера, и не имеет возможности узнать промежуточный результат этих инструкций, пока они не будут выполнены окончательно.

Пусть дети нарисуют картинки и запишут свои собственные инструкции. Попробуйте организовать пары или задействовать весь класс.

## **Вариации**

1. Напишите инструкции для построения бумажного копья (дротика для метания или самолетика и др.).
2. Напишите инструкции для нахождения тайника при обходе школы с использованием таких инструкций, как «Идите вперед  $x$  метров», «Поворот налево» (90 градусов), и «поворот направо» (90 градусов).

Дети должны проверять и уточнять указания, пока получат желаемого результата.

3. Слепая игра. С завязанными глазами направляете по комнате одного ребенка или группу детей.

# О чём это всё?

---

Компьютеры работают в соответствии с определенным набором инструкций, называемым программой, разработанной для конкретной задачи. Программы написаны на языках, с ограниченным набором инструкций, указывающих компьютерам выполняемые команды. Определенные языки подходят более для одних целей и задач, и менее для других.

Независимо от того, какой язык используется, программисты должны иметь большой опыт в написании *точных* указаний компьютеру, что они хотят, чтобы компьютер выполнил. В отличие от человека, компьютер будет выполнять письменные инструкции, даже если они явно смешны.



Важно, чтобы программа была хорошо написана. Небольшая ошибка может вызвать много проблем. Представьте себе последствия ошибки в компьютерной программе для запуска космического челнока, атомной электростанции, или сигналов на пути следования поезда! Ошибки обычно называются “bugs”, в честь (так сказано) жуков, что когда-то были удалены ((“debugged”) (“отлаживать”)) с электрического реле электронной вычислительной машины в начале 1940-х.

В более сложных программах, вероятность большего количества ошибок увеличивается. Это стало серьезной проблемой, когда США стали работать над программой Стратегических Оборонных Инициатив (“Звездные войны”), с системой компьютерного управления, которая была предназначена для формирования непроницаемой защиты от ядерного удара. Некоторые ученые утверждали, что им трудно работать из-за сложности и ненадежности программного обеспечения. Программное обеспечение должно быть тщательно проверено, и найдено как можно больше ошибок, чтобы не проверять её надежность на практике, непосредственно при обстреле ракетами США!