ИвГУ, ф-т МиКН, курс 2

"КОМПЬЮТЕРНАЯ АЛГЕБРА"

Тема 0.

Что такое компьютерная алгебра?

Лектор: Н. И. Яцкин, 2014

АЛГЕБРА

АЛГОРИТМЫ

Точно работающие

Приближенные

КОМПЬЮТЕРНАЯ АЛГЕБРА

Численные методы алгебры

Два возможных направления работы:

(1) изучение алгебраических объектов, понятий и методов, применяемых в компьютерных науках;

(2) изучение компьютерных программных продуктов, применяемых в математике.

"Компьютерная алгебра рассматривает такие объекты,
которые имеют слишком вычислительный характер,
чтобы встречаться в книгах по алгебре,
и слишком алгебраический характер,
чтобы быть представленными в учебниках по информатике."

Компьютерная алгебра:
Символьные и алгебраические вычисления.
Б. Бухбергер и др. (ред.)
см. список рекомендуемой литературы

Что понадобится из математики?

1. **Теория чисел** (алгоритм Евклида, делимость, сравнения, простые числа, разложение на простые множители).
2. **Абстрактная алгебра** (группы, кольца, поля).
3. **Алгебра многочленов** (алгоритм Евклида, неприводимые многочлены, разложение на неприводимые множители).
4. **Линейная алгебра** (матрицы, системы линейных уравнений, алгоритм Гаусса, линейные пространства, базисы).

Чего (к сожалению) не будет в нашем курсе?

1. *Способы компьютерного представления*

ОЧЕНЬ БОЛЬШИХ

натуральных чисел (и т. п. вопросы).

Целое число многократной точности можно рассматривать как число в системе счисления по основанию b,
где b - максимальное число однократной точности.

2. *Понятие о*

сложности алгоритмов

и методах ее оценки.

Согласно 

:

Временная сложность алгоритма (в худшем случае) — это функция размера входных данных, равная максимальному количеству элементарных операций, проделываемых алгоритмом для решения экземпляра задачи указанного размера.

Точнее это выражается с помощью асимптотических обозначений типа

"О большое".

Что вы помните из математического анализа про "О большое"?

Алгоритм имеет ***сложность***

$$t(n) = O(f(n)),$$

если при увеличении ***размерности*** *входных данных* n,

время $t(n)$ *выполнения алгоритма возрастает*

с не большей скоростью, чем (положительная) функция $f(n)$,

т. е. существует такая (положительная) константа C,

что при $n \to \infty$:

$$t(n) \le C\, f(n).$$

Пример. ***Полиномиальная сложность:***

$$t(n) = O(n^s);$$

где s – натуральное число.

Другой подход: ***сложность - это***

математическое ожидание

времени работы алгоритма.

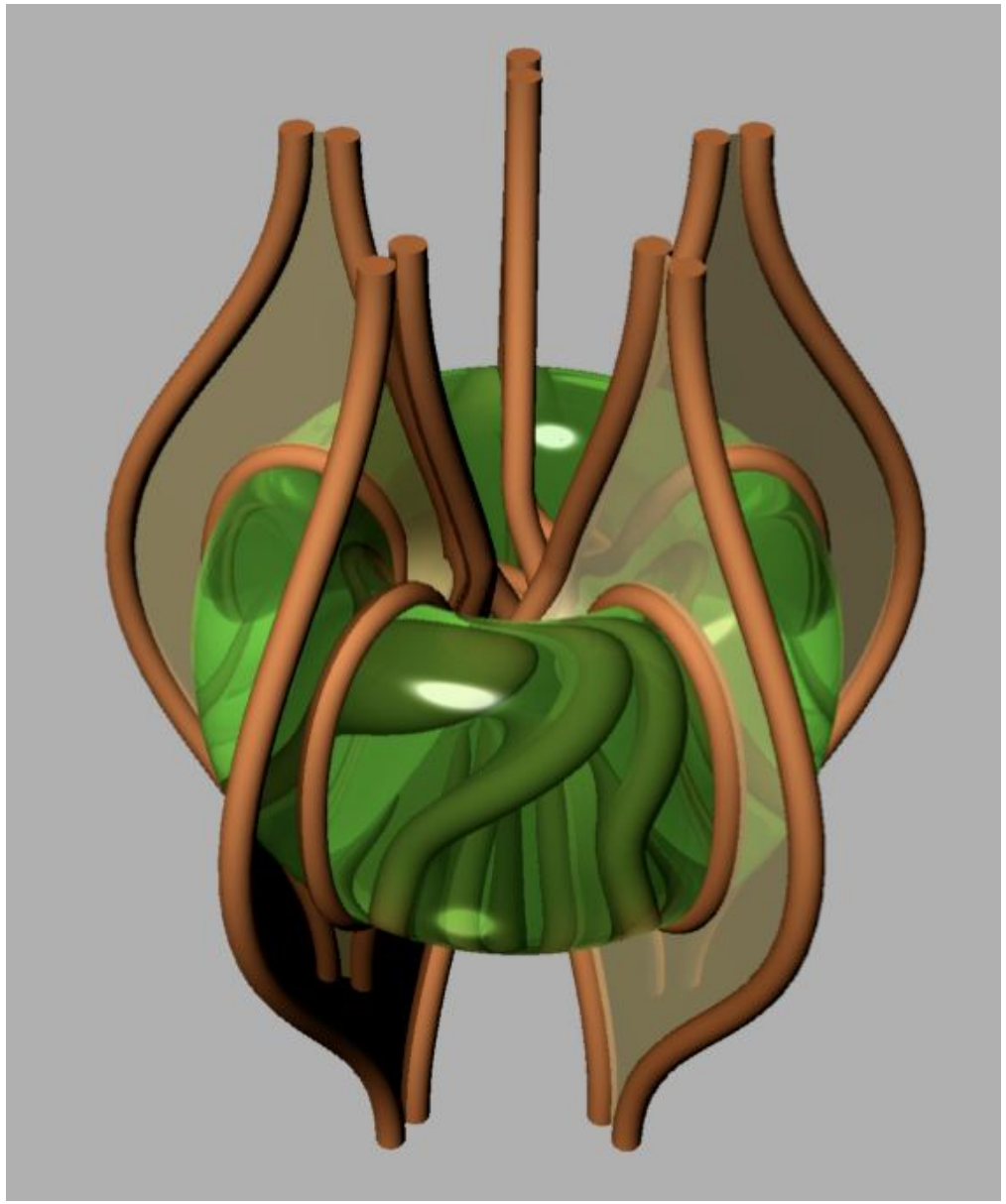
Что вам известно из теории вероятностей
про "математическое ожидание"?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. **Акритас А.** ***Основы компьютерной алгебры с приложениями.*** М.: Мир, **1994**. 544 с.
2. **Дэвенпорт Д., Сирэ И., Турнье Э.** ***Компьютерная алгебра***. М.: Мир, **1991**. 352 с.
3. **Панкратьев Е. В.** ***Элементы компьютерной алгебры.*** М.: БИНОМ. Лаборатория знаний, **2007.** 248 с.
4. **Василенко О.Н.** ***Теоретико-числовые алгоритмы в криптографии***. М.: Изд-во МЦНМО, **2003.** 326 с.
5. **Гашков С. В., Чубариков В. Н**. ***Арифметика. Алгоритмы. Сложность вычислений***. М.: Изд-во МГУ, **2009.** 320 с.
6. **Кокс Д., Литтл Дж., О'Ши Д.** ***Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры.*** М.: Мир, **2000.** 688 с.
7. ***Компьютерная алгебра. Символьные и алгебраические вычисления*** / Под ред. Б. Бухбергера, Д. Коллинза, Р. Лооса. М..: Мир, **1986**. 391 с.
8. **Ноден П., Китте К.** ***Алгебраическая алгоритмика. С упражнениями и решениями***. М.: Мир, **1999.** 720 с.

ДОПОЛНИТЕЛЬНЫЙ СПИСОК

1. **Берлекэмп Э.** ***Алгебраическая теория кодирования.*** М.: Мир, **1971**. 479 с.
2. **Коблиц Н.** ***Курс теории чисел и криптографии.*** М.: Научное издательство "ТВП", **2001.** 260 с.
3. **Коутинхо С.** ***Введение в теорию чисел. Алгоритм RSA***. М.: ПОСТМАРКЕТ, **2001**. 212 с.
4. **Смарт Н.** ***Криптография.*** М.: Техносфера, **2005.** 528 с.
5. **Черемушкин А. В.** ***Лекции по арифметическим алгоритмам в криптографии***. М.: МЦМНО, **2002**. 104 с.



CAS

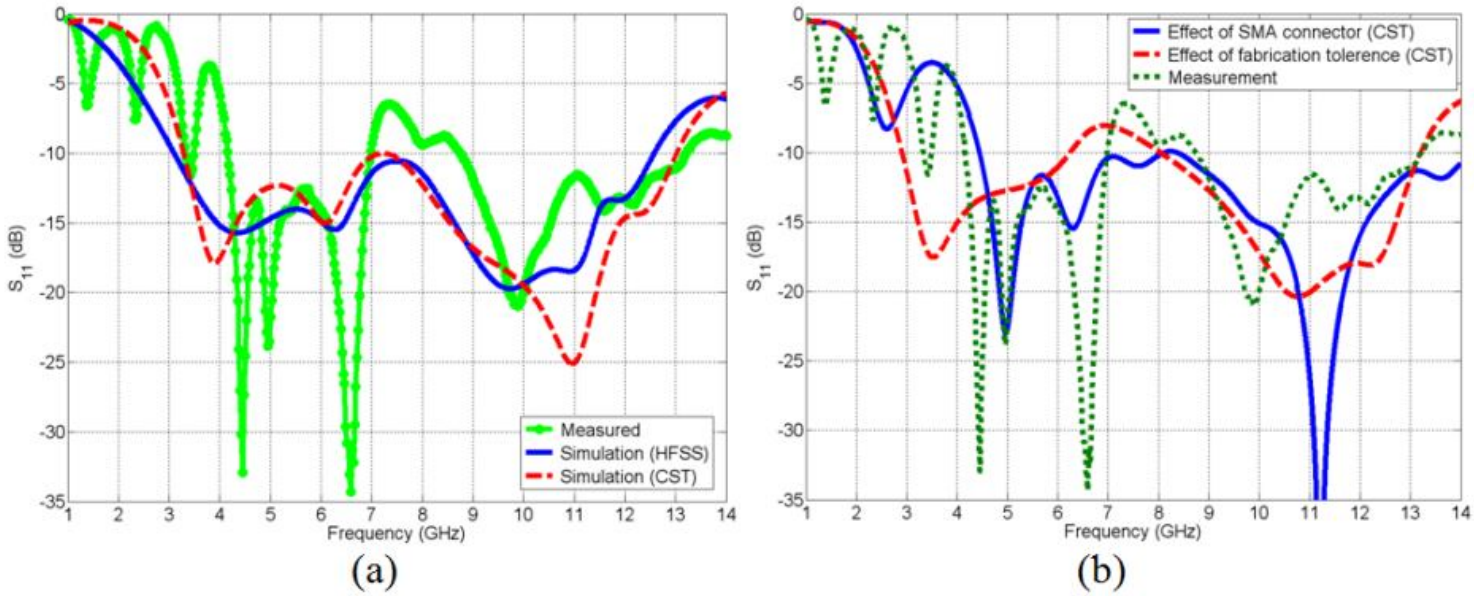
Computer Algebra Systems

Система компьютерной алгебры (***computer algebra system, CAS***)
- программное приложение для **символьных вычислений**,
т. е. для выполнения преобразований и работы с математическими выражениями в аналитической (символьной) форме.

Обычно CAS поддерживают следующие символьные действия:

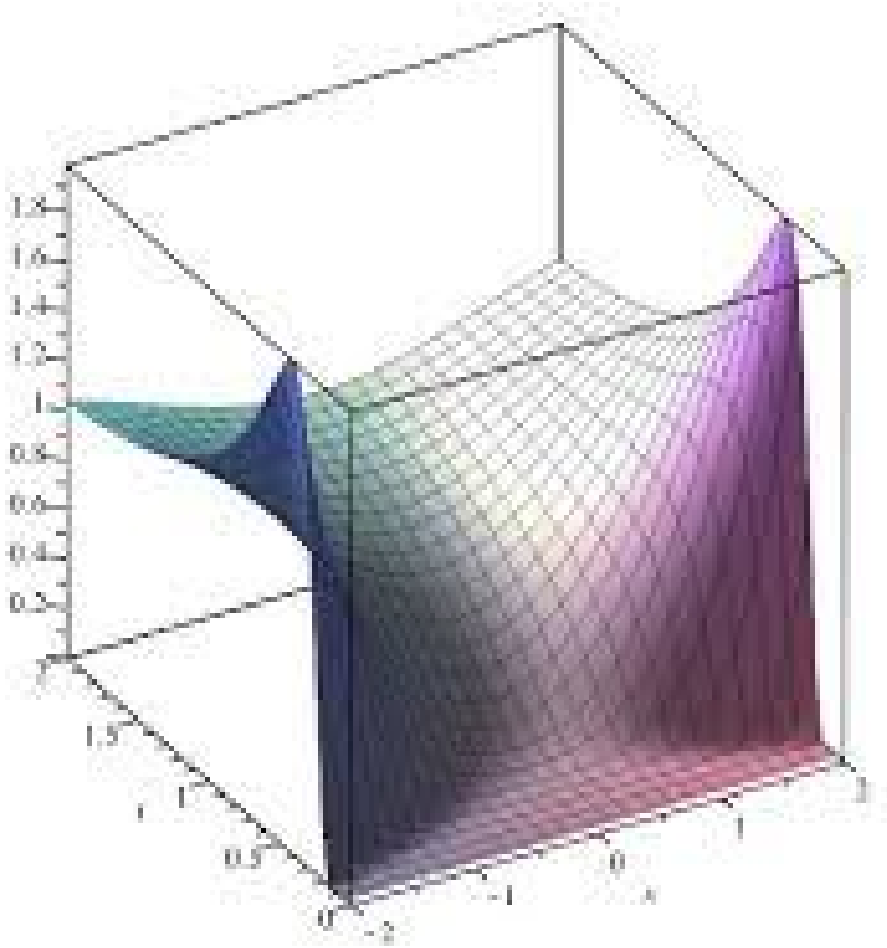
- **упрощение выражений** до меньшего размера или приведение к стандартному виду;
- **подстановка** символьных и численных значений в выражения;
- изменение вида выражений: раскрытие произведений и степеней, частичная и полная **факторизация** (разложение на множители);

- дифференцирование;
- нахождение неопределённых и определённых интегралов (**символьное интегрирование**);
- **решение** линейных и нелинейных **уравнений**;
- алгебраическое (нечисленное) **решение дифференциальных уравнений**;
- **нахождение пределов** функций и последовательностей;
- **оперирование с рядами**: суммирование, умножение, суперпозиция;
- **матричные операции**: обращение, факторизация, решение спектральных задач
- **операции со строками**.



Дополнительные возможности:

- **язык программирования**;
- **числовые операции** произвольной точности;
- целочисленная **арифметика больших чисел**, поддержка теоретикочисловых функций;
- **построение графиков** функций (**2d** и **3d**), **анимация**.



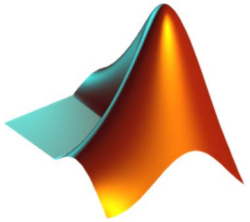
История: Появились в начале 1960-х.

Лидеры:

Maple, MATLAB, Mathematica

Конкуренты (свободно распространяемые)

Sage, GAP, Maxima, Axiom




Maple

Является продуктом компании **Waterloo Maple Inc.**, выпускает программные продукты, ориентированные на сложные математические вычисления, визуализацию данных и моделирование. Предназначена для ***символьных*** вычислений; имеет ряд средств для ***численного*** решения задач; обладает развитыми ***графическими*** средствами; имеет собственный язык программирования, напоминающий Паскаль.

http://www.maplesoft.com/products/Maple/index.aspx

Waterloo Maple Inc. was first incorporated under the name
Waterloo Maple Software in April 1988
in the **Symbolic Computation Group**,
a part of the **computer science department** at the **University of Waterloo**.
http://uwaterloo.ca/ https://cs.uwaterloo.ca/

University of Waterloo (Canada)

University of Ivanovo (Russia)

Версии:

- Maple 17 13 марта 2013
- Maple 16 28 марта 2012
- Maple 15 13 апреля 2011
- Maple 14 29 апреля 2010
- Maple 13 24 апреля 2009
- Maple 12 13 мая 2008
- **********
- Maple V: август 1990
- **********
- Maple 1.0: январь 1982

Что значит "аналитическое решение уравнений"?

Численное решение квадратного уравнения $x^2 + x - 1 = 0$:

$$x_1, x_2 = 0.61803, -1.61803$$

КАС ищет **точное** (символьное) решение:

```
> x[1],x[2]:=solve(x^2+x-1=0);
```

$$x_1, x_2 := -\frac{1}{2} + \frac{\sqrt{5}}{2}, -\frac{1}{2} - \frac{\sqrt{5}}{2}$$

Но может перейти и к **численным** (приближенным) ответам:

```
> x[1],x[2]:=evalf(x[1]),evalf(x[2]);
```

$$x_1, x_2 := 0.6180339880, -1.618033988$$

Символьное решение уравнения **пятой** степени

$$x^5 + 2x^4 + x^2 + x - 3 = 0$$

невозможно ввиду несуществования аналитических формул для корней уравнений степени, большей четырех.

Однако корни (***в поле комплексных чисел***) существуют, их пять штук (с учетом кратностей), и **КАС** умеет (с помощью специальной функции **RootOf**) производить над ними различные действия так, как будто они известны точно, получая при этом точные результаты.

Вот сами корни (пока только **"зарегистрированные"** системой):

```
> x[1],x[2],x[3],x[4],x[5]:=solve(x^5+2*x^4+x^2-3=0);
```

$$x_1, x_2, x_3, x_4, x_5 := \mathrm{RootOf}(\_Z^5 + 2\,\_Z^4 + \_Z^2 - 3, index = 1),$$

$$\mathrm{RootOf}(\_Z^5 + 2\,\_Z^4 + \_Z^2 - 3, index = 2), \mathrm{RootOf}(\_Z^5 + 2\,\_Z^4 + \_Z^2 - 3, index = 3),$$

$$\mathrm{RootOf}(\_Z^5 + 2\,\_Z^4 + \_Z^2 - 3, index = 4), \mathrm{RootOf}(\_Z^5 + 2\,\_Z^4 + \_Z^2 - 3, index = 5)$$

Вычислим сумму и произведение корней:

```
> evala(x[1]+x[2]+x[3]+x[4]+x[5]);
                                -2

> evala(x[1]*x[2]*x[3]*x[4]*x[5]);
                                3
```

Полученные результаты согласуются с **теоремой Виета**.

Вычислим приближенные значения корней (символом ***I*** обозначается мнимая единица):

```
> x[1],x[2],x[3],x[4],x[5]:=
                  evalf(x[1]),evalf(x[2]),evalf(x[3]),
                  evalf(x[4]),evalf(x[5]);
```

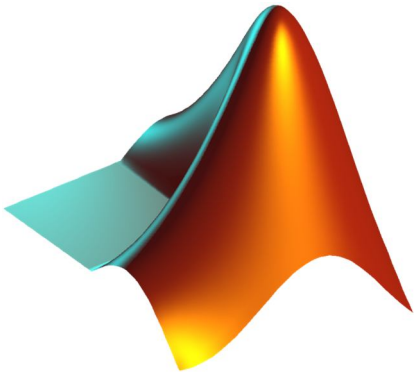
$$x_1, x_2, x_3, x_4, x_5 := 0.9252531992,\ 0.1638945660 + 1.138918544\ I,\ -1.183061834,$$
$$-2.069980497,\ 0.1638945660 - 1.138918544\ I$$

Снова найдем их произведение:

```
> evalf(x[1]*x[2]*x[3]*x[4]*x[5]);
```

$$2.999999999 - 0.394167100\ 10^{-10}\ I$$

Результат отличается от точного ответа в десятом знаке после десятичной точки.

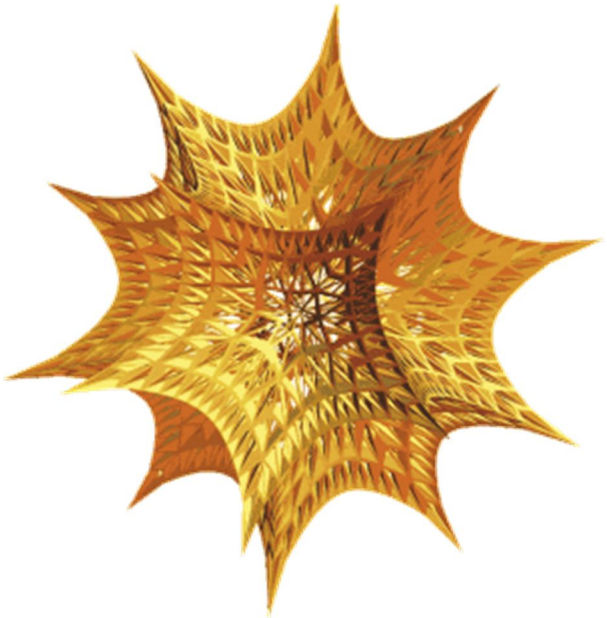


MATLAB

Matrix Laboratory

- пакет прикладных программ
для решения задач технических вычислений
и одноимённый язык программирования.

http://www.mathworks.com/products/matlab/



Mathematica

- система компьютерной алгебры компании Wolfram Research.

http://www.wolfram.com/

Sage (*мудрец*) — 

первая версия выпущена 24 февраля 2005 года в виде свободного программного обеспечения.

Цель проекта: ***создание открытого программного обеспечения альтернативного системам*** **`Magma, Maple, Mathematica`**, *и* **MATLAB**.

http://www.sagemath.org/

Философия разработки Sage

- Для создания достойной альтернативы системам **Magma, Maple, Mathematica**, и **MATLAB** потребуются сотни или тысячи человеко-лет, если начинать процесс разработки с нуля.
- Существует большое количество готового математического ПО с открытым исходным кодом, но написанного на различных языках программирования, из которых наиболее встречаемыми являются **C, C++, Fortran** и **Python**.

Таким образом, вместо того, чтобы начинать с нуля, было решено объединить всё специализированное математическое ПО в систему с общим интерфейсом.

Конечному пользователю необходимо лишь знать язык **Python**.

Если для какой-то частной задачи не существовало ПО с открытым кодом, тогда стояла задача написания соответствующего блока для Sage. К разработке Sage привлекаются как профессионалы, так и студенты. Разработчики работают на общественных началах и поддерживаются грантами.

Microsoft спонсировала разработку версии Sage специально под ОС Windows, но на данный момент пользователям этой операционной системы нужно использовать технологию *виртуализации* для работы с Sage. Рекомендуемая программа виртуализации — VirtualBox

Включает в себя системы (пакеты):

GAP, Maxima, Singular

GAP

http://ru.wikipedia.org/

GAP (от англ. ***Groups, Algorithms, Programming***) — свободно распространяемая кроссплатформенная система компьютерной алгебры для вычислительной дискретной алгебры с особым вниманием к ***вычислительной теории групп***.

Совместная разработка университетов
Сент-Эндрюс (Шотландия),
Ахен, Брауншвейг (Германия),
Колорадо (США).

Возможности системы GAP можно расширить используя внешние пакеты и библиотеки, либо воспользовавшись ***паскалеподобным языком программирования***, также называемым **GAP**.

Пример кода:

```
FixOrdElms:=function(G, k)
  local L;
L:=AsList(G);
L:=Filtered(L,x->Order(x)=k);
return L;
end;
```

GAP-язык включает идентификаторы для таких сложных объектов как ***группы, кольца, поля***.

Описание функции:

FixOrdElms - функция от двух аргументов; первым аргументом принимает *группу* **G**, вторым – ***натуральное число*** **k**.

Функция **AsList** возвращает все элементы группы **G**, в форме ***списка*****.** К списку затем применяет ***фильтр***, отбирающий элементы заданного порядка **k**. Порядок элемента определяется с помощью функции **Order**.

Примеры работы:

```
FixOrdElms(SymmetricGroup(5),3);
```

[(3,4,5), (3,5,4), (2,3,4), (2,3,5), (2,4,3), (2,4,5), (2,5,3), (2,5,4), (1,2,3), (1,2,4), (1,2,5), (1,3,2), (1,3,4), (1,3,5), (1,4,2), (1,4,3), (1,4,5), (1,5,2), (1,5,3), (1,5,4)]

```
FixOrdElms(AlternatingGroup(4),2);
```

[(1,2)(3,4), (1,3)(2,4), (1,4)(2,3)]

ИвГУ, ф-т МиКН, 2012/13 уч.г.,2-й курс,
дисциплина "Компьютерная алгебра",
студенты ***А. Куваев, А. Смоляков,***
методическое пособие

Система компьютерной алгебры SAGE: установка и основы программирования.
Изд-во "Ивановский гос. ун-т", 2013. 35 с.

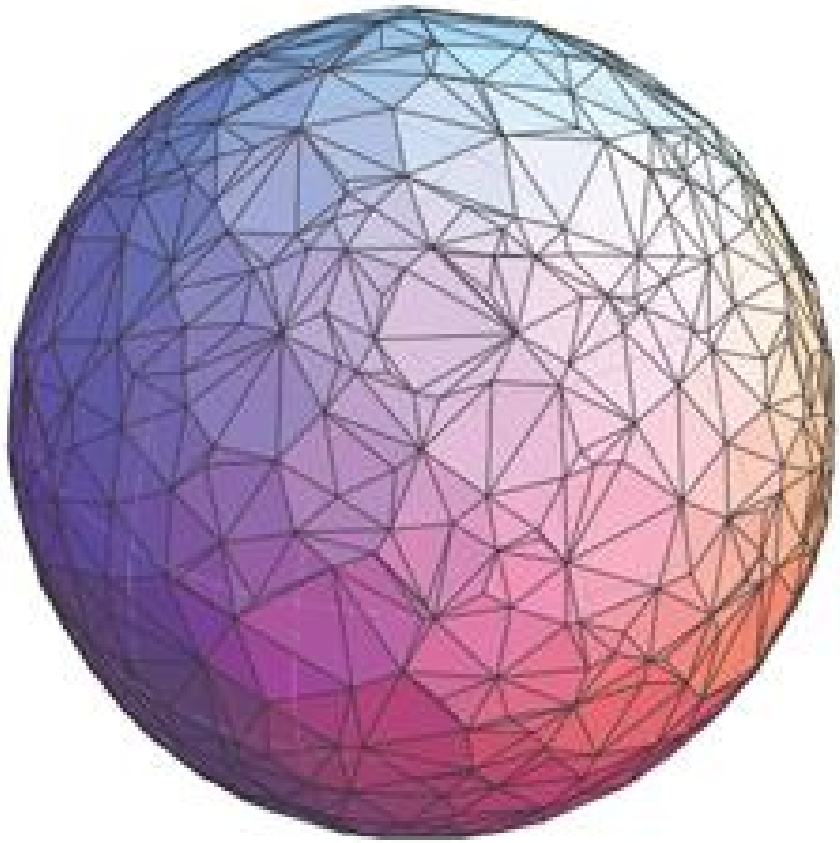
http://lib.ivanovo.ac.ru:81/elib/dl/matematika/metod/kuvaev_2014.pdf/

Александр Куваев (ИвГУ)

Подготовлено продолжение:

Яцкин Н. И. Алгебраические вычисления в системе SAGE.

ИвГУ, 2014. 47 с.



Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.

Узнайте этого человека,
прочитайте и переведите
его замечательную максиму.

Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.

Бог создал целые числа,
всё остальное —
дело рук человека.
Л. Кронекер

Первичным объектом во всех системах компьютерной алгебры является ***полукольцо***

$$(\mathbf{N}; +, \cdot, \mathbf{0}, \mathbf{1})$$

натуральных чисел.

Кстати, знаете ли вы, чему равна сумма всех натуральных чисел?

Некоторые считают, что

$$\sum_{n=1}^{\infty} n = -\frac{1}{12}$$

. . . . . .

Теперь вопросы попроще.

(1) Дайте определение простого натурального числа.

(2) Докажите бесконечность множества простых натуральных чисел.

(3) Сформулируйте основную теорему арифметики.

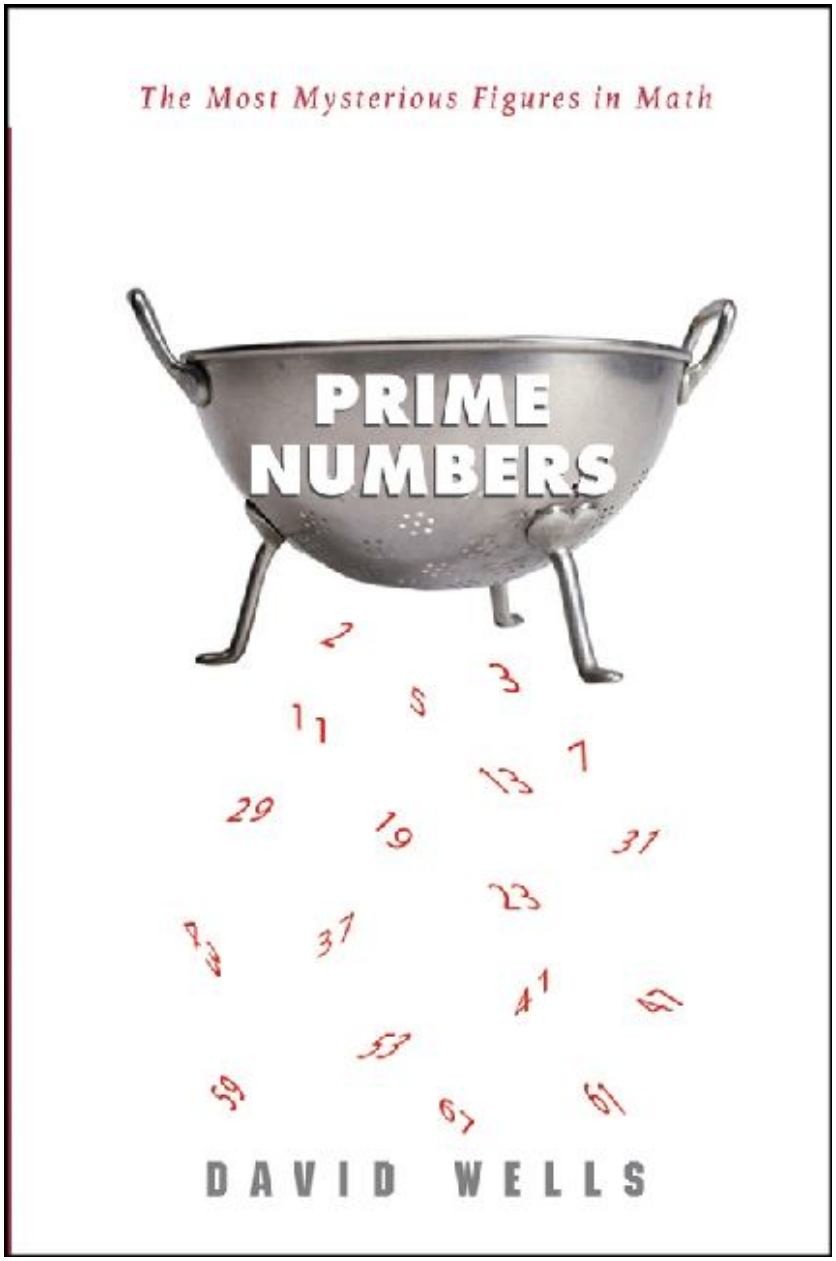
David Wells.

PRIME NUMBERS.

The Most Mysterious Figures in Math.

John Wiley & Sons, Inc.,

2005.



РЕКОРДЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

Числа Мерсе́нна — натуральные числа вида

$$M_n = 2^n - 1,$$

где n — натуральное число. Если M_n – простое, то n — тоже простое. До 2013 года было известно **47** чисел Мерсенна.
В январе 2013 открыто **48**-е:

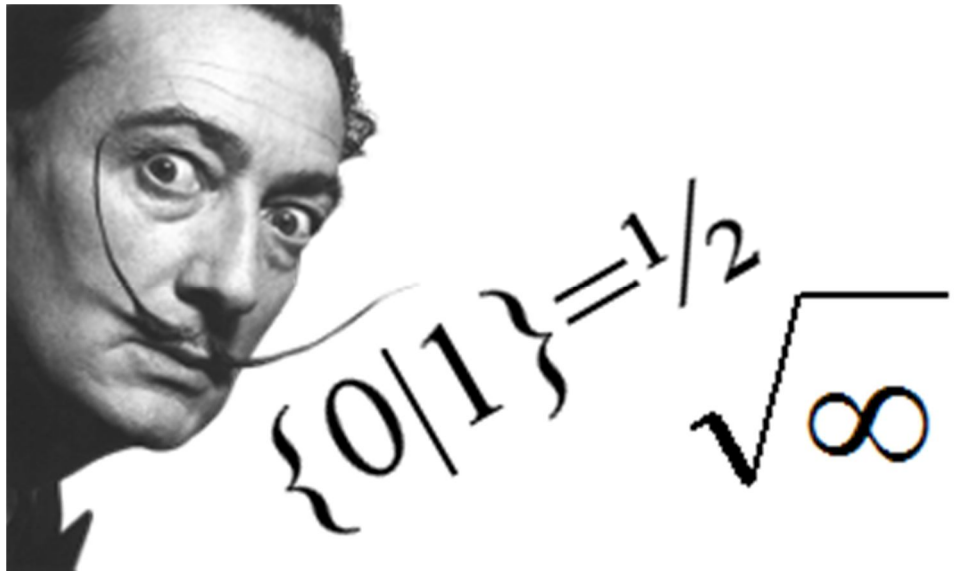
$$M_{57\,885\,161} = 2^{57\,885\,161} - 1.$$

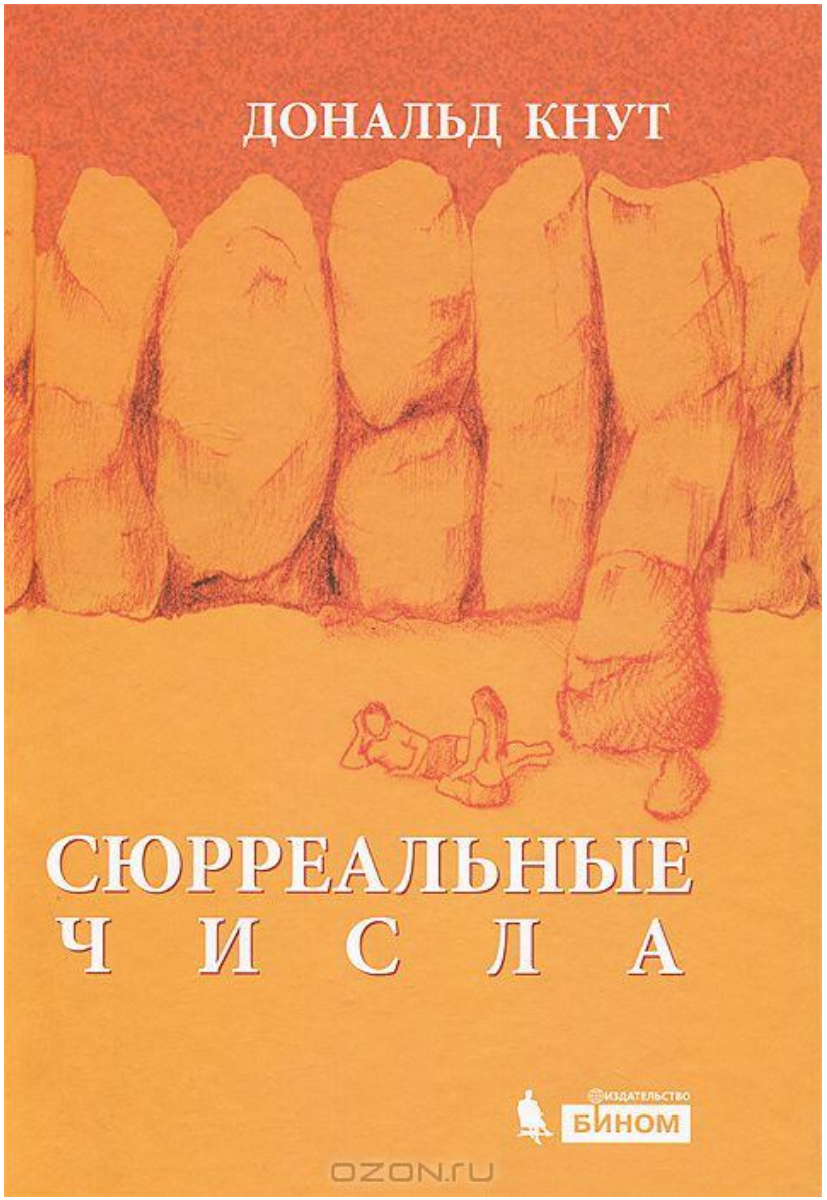
(17 425 170 *digits*)

А есть еще реальные (действительные) и сюрреальные числа.

Величайший программист, мудрец Donald Knuth …

…придумал это название.

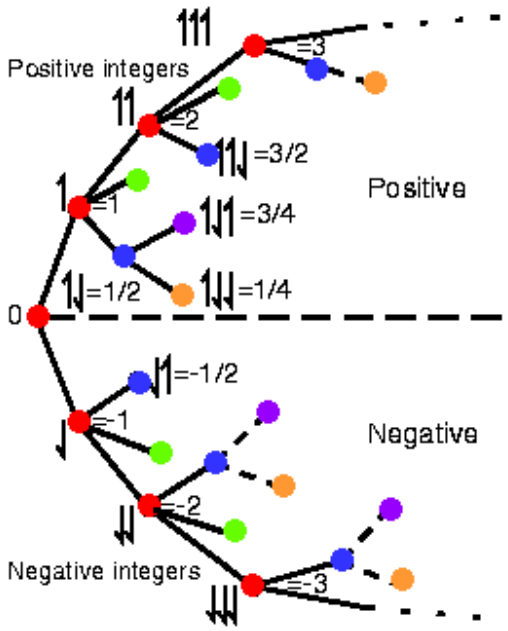





"Computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty."
Donald Knuth 1974

Сами числа изобрел другой мудрец,

John Conway.



Но…

ПЕРЕХОДИМ К ДЕЛУ!

ПЕРЕХОДИМ К ДЕЛУ!

ПЕРЕХОДИМ К ДЕЛУ!

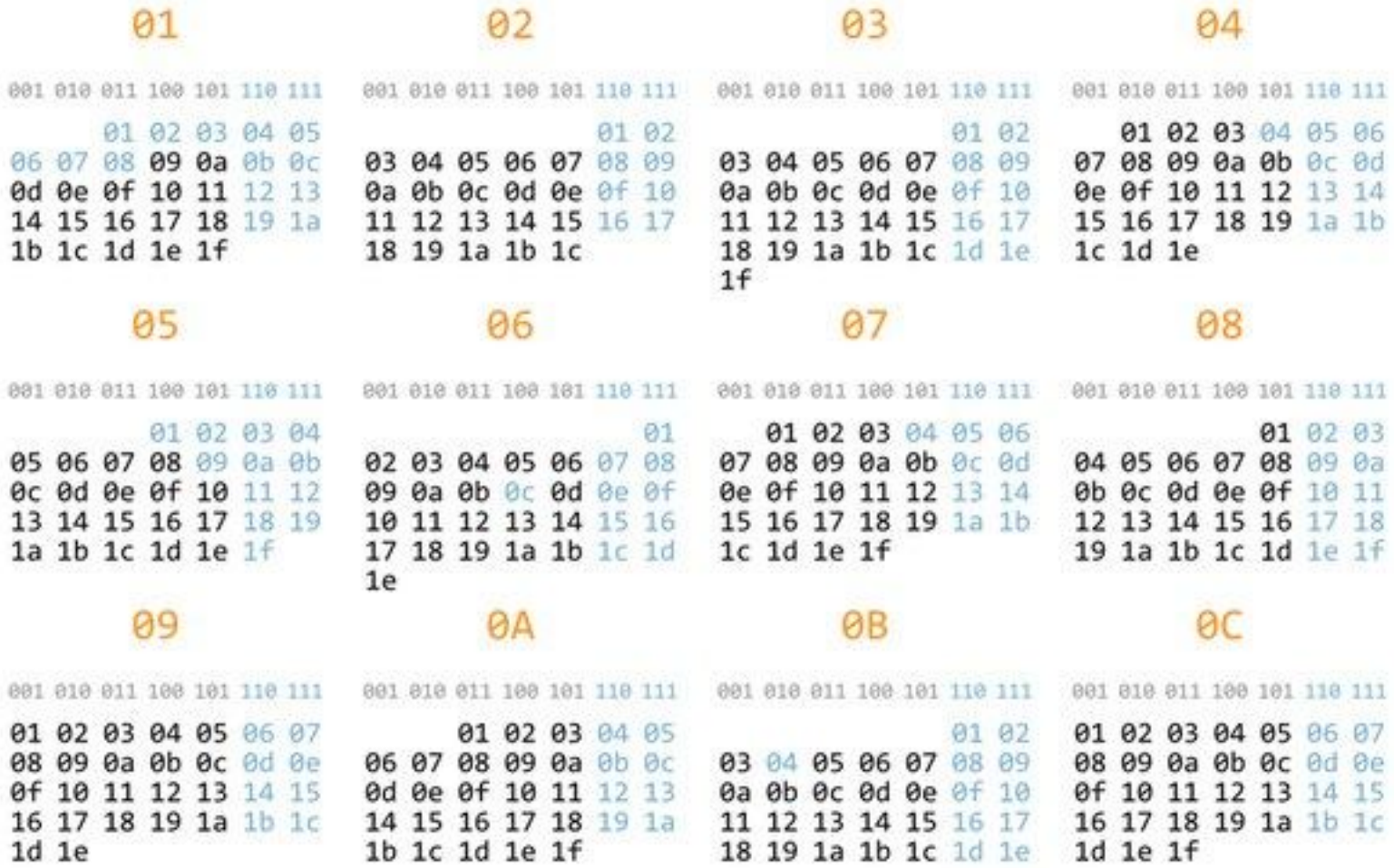
ПЕРЕХОДИМ К ДЕЛУ!

ПЕРЕХОДИМ К ДЕЛУ!

ПЕРЕХОДИМ К ДЕЛУ!

ПЕРЕХОДИМ К ДЕЛУ!

0x7DE


01
001 010 011 100 101 110 111
01 02 03 04 05
06 07 08 09 0a 0b 0c
0d 0e 0f 10 11 12 13
14 15 16 17 18 19 1a
1b 1c 1d 1e 1f
02
001 010 011 100 101 110 111
01 02
03 04 05 06 07 08 09
0a 0b 0c 0d 0e 0f 10
11 12 13 14 15 16 17
18 19 1a 1b 1c
03
001 010 011 100 101 110 111
01 02
03 04 05 06 07 08 09
0a 0b 0c 0d 0e 0f 10
11 12 13 14 15 16 17
18 19 1a 1b 1c 1d 1e
1f
04
001 010 011 100 101 110 111
01 02 03 04 05 06
07 08 09 0a 0b 0c 0d
0e 0f 10 11 12 13 14
15 16 17 18 19 1a 1b
1c 1d 1e
05
001 010 011 100 101 110 111
01 02 03 04
05 06 07 08 09 0a 0b
0c 0d 0e 0f 10 11 12
13 14 15 16 17 18 19
1a 1b 1c 1d 1e 1f
06
001 010 011 100 101 110 111
01
02 03 04 05 06 07 08
09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16
17 18 19 1a 1b 1c 1d
1e
07
001 010 011 100 101 110 111
01 02 03 04 05 06
07 08 09 0a 0b 0c 0d
0e 0f 10 11 12 13 14
15 16 17 18 19 1a 1b
1c 1d 1e 1f
08
001 010 011 100 101 110 111
01 02 03
04 05 06 07 08 09 0a
0b 0c 0d 0e 0f 10 11
12 13 14 15 16 17 18
19 1a 1b 1c 1d 1e 1f
09
001 010 011 100 101 110 111
01 02 03 04 05 06 07
08 09 0a 0b 0c 0d 0e
0f 10 11 12 13 14 15
16 17 18 19 1a 1b 1c
1d 1e
0A
001 010 011 100 101 110 111
01 02 03 04 05
06 07 08 09 0a 0b 0c
0d 0e 0f 10 11 12 13
14 15 16 17 18 19 1a
1b 1c 1d 1e 1f
0B
001 010 011 100 101 110 111
01 02
03 04 05 06 07 08 09
0a 0b 0c 0d 0e 0f 10
11 12 13 14 15 16 17
18 19 1a 1b 1c 1d 1e
0C
001 010 011 100 101 110 111
01 02 03 04 05 06 07
08 09 0a 0b 0c 0d 0e
0f 10 11 12 13 14 15
16 17 18 19 1a 1b 1c
1d 1e 1f