МГУ – ШКОЛЕ

Московский государственный университет
имени М.В. Ломоносова

Механико-математический факультет

Университетская суббота

Современная математика: от основ к искусственному интеллекту

Самоненко Илья Юрьевич

20 сентября 2014 года

Университетские субботы

Всего в Московском государственном университете имени М.В.Ломоносова состоится около 30 Университетских суббот.

Сегодня проходят два мероприятия – на механико-математическом факультете и в Музее землеведения.

Университетские субботы

Всего в Московском государственном университете имени М.В.Ломоносова состоится около 30 Университетских суббот.

Сегодня проходят два мероприятия – на механико-математическом факультете и в Музее землеведения.

Основная цель – познакомить с разнообразием научных направлений, помочь Вам с выбором будущего образования.

Университетские субботы

Всего в Московском государственном университете имени М.В.Ломоносова состоится около 30 Университетских суббот.

Сегодня проходят два мероприятия – на механико-математическом факультете и в Музее землеведения.

Основная цель – познакомить с разнообразием научных направлений, помочь Вам с выбором будущего образования.

Полтора часа – не так много времени…
задача в большей степени удивить, чем научить чему-то…

О чем наша лекция?

В лекции будет 4 части.

О чем наша лекция?

В лекции будет 4 части.

Сформулируем несколько провокационно:

Часть 1. О разнообразии математических миров.
Часть 2. Компьютер ничего не может…
Часть 3. Компьютер что-то может …
Часть 4. Компьютер может все…

О чем наша лекция?

В лекции будет 4 части.

Сформулируем несколько провокационно:

Часть 1. О разнообразии математических миров.
Часть 2. Компьютер ничего не может… без человека.
Часть 3. Компьютер что-то может … при помощи человека.
Часть 4. Компьютер может все… что сделает на нем человек.

О чем наша лекция?

В лекции будет 4 части.

Сформулируем несколько провокационно:

Часть 1. О разнообразии математических миров.
Часть 2. Компьютер ничего не может… без человека.
Часть 3. Компьютер что-то может … при помощи человека.
Часть 4. Компьютер может все… что сделает на нем человек.

Сформулируем более строго:

Часть 1. Дискретная, непрерывная и фрактальная математика.
Часть 2. Алгоритмически неразрешимые задачи.
Часть 3. Трунорешаемые задачи и проблема P=NP.
Часть 4. Биологические алгоритмы.

Математические миры

Математика

Дискретная
- построена из «кирпичиков»

Арифметика с целыми числами

□□ + □□□ = □□□□□

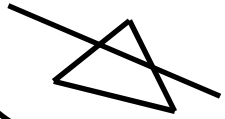
Математическая логика

*Если из **А** следует **В**,*
*то из **не В** следует **не А***

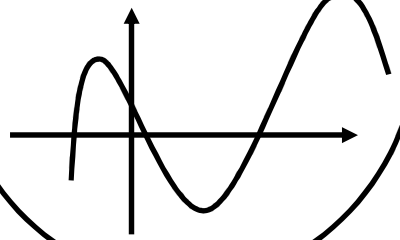
и др.

«Непрерывная» -
Построена из бесконечных, «непрерывных» объектов

Геометрия



Математический анализ



и др.

Математические миры

Математика

Дискретная
- построена из «кирпичиков»

Арифметика с целыми числами

□□ + □□□ = □□□□□

Математическая логика

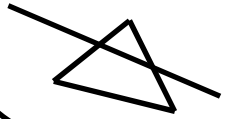
*Если из **А** следует **В**,*
*то из **не В** следует **не А***

Наша лекция об этом

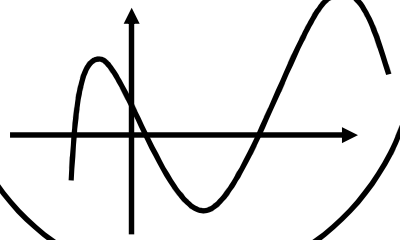
и др.

«Непрерывная» -
Построена из бесконечных, «непрерывных» объектов

Геометрия



Математический анализ

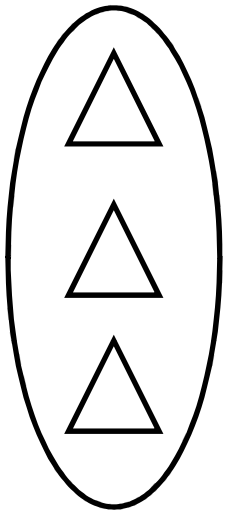


и др.

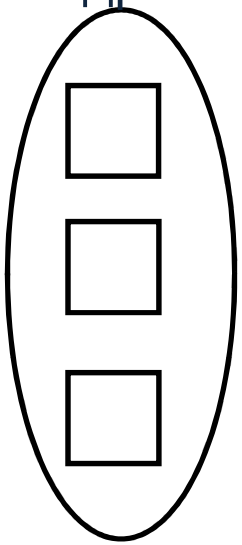
Дискретная и непрерывная. В чем разница?

Дискретная

Множество треугольников



Множество квадратов



В каком множестве больше элементов?

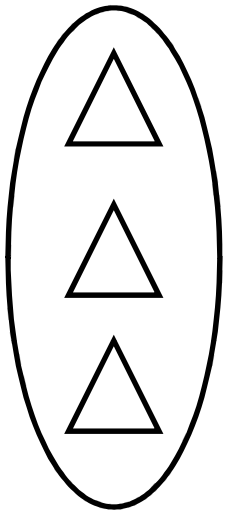
Непрерывная

Дискретная и непрерывная. В чем разница?

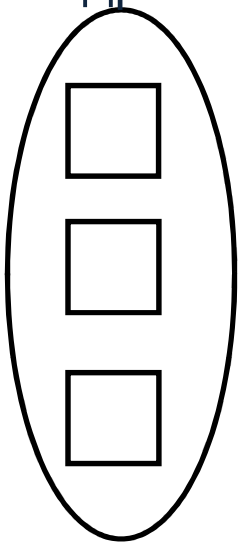
МГУ – ШКОЛЕ

Дискретная

Множество треугольников



Множество квадратов



В каком множестве больше элементов?

Одинаково!
Как мы это поняли?

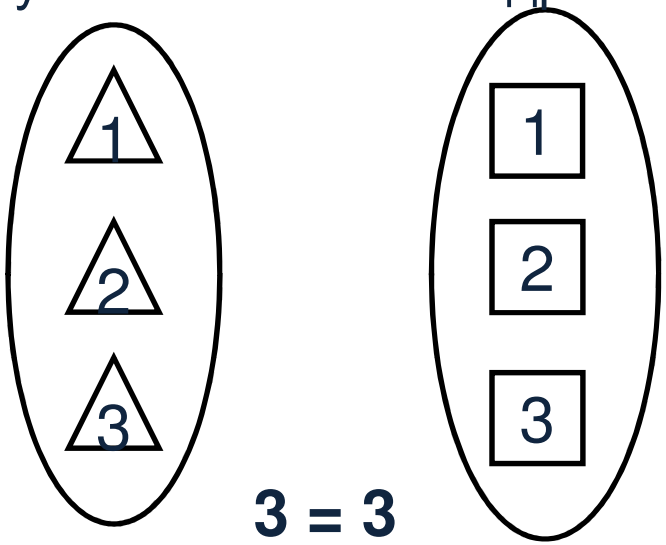
Непрерывная

Дискретная и непрерывная. В чем разница?

Дискретная

Множество треугольников

Множество квадратов



3 = 3

В каком множестве больше элементов?

Одинаково!
Как мы это поняли?

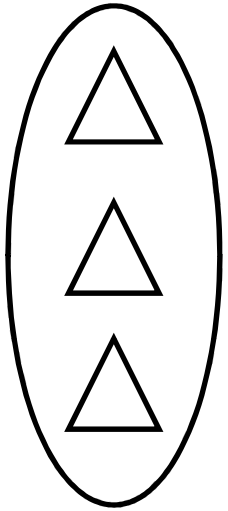
Пересчитали!

Непрерывная

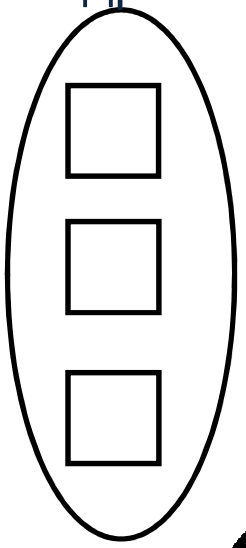
Дискретная и непрерывная. В чем разница?

Дискретная

Множество треугольников



Множество квадратов



Разучился считать до трех… Как быть?

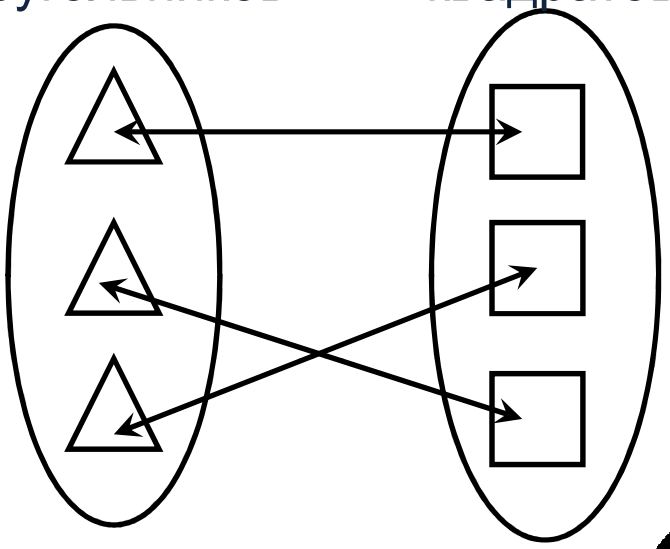
Непрерывная

Дискретная и непрерывная. В чем разница?

Дискретная

Множество треугольников

Множество квадратов



Разучился считать до трех… Как быть?

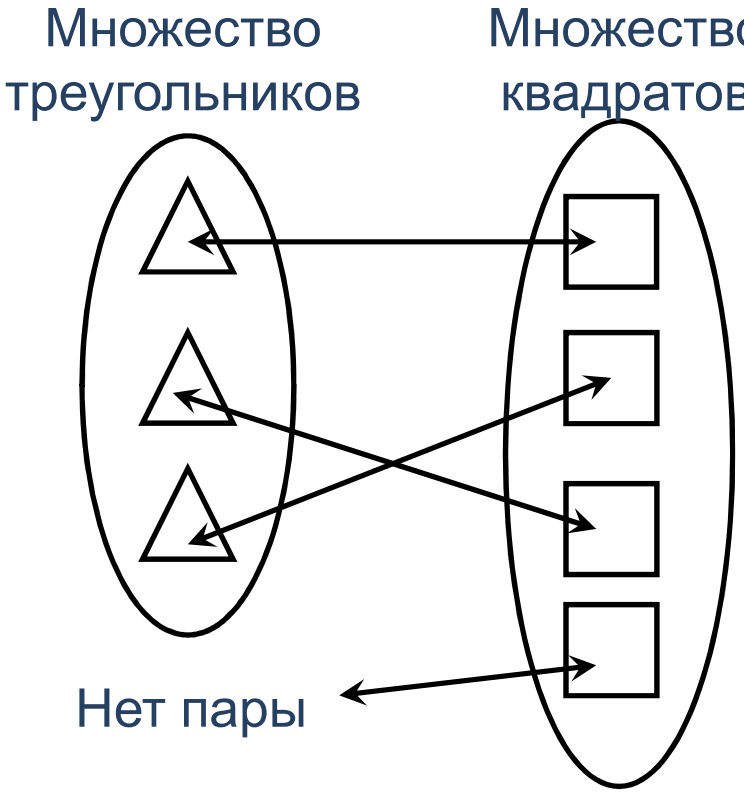
Установили **взаимно-однозначное соответствие** между элементами двух множеств. Следовательно в данных множествах **одинаковое число элементов**.

Множества **равномощны**.

Непрерывная

Дискретная и непрерывная. В чем разница?

Дискретная



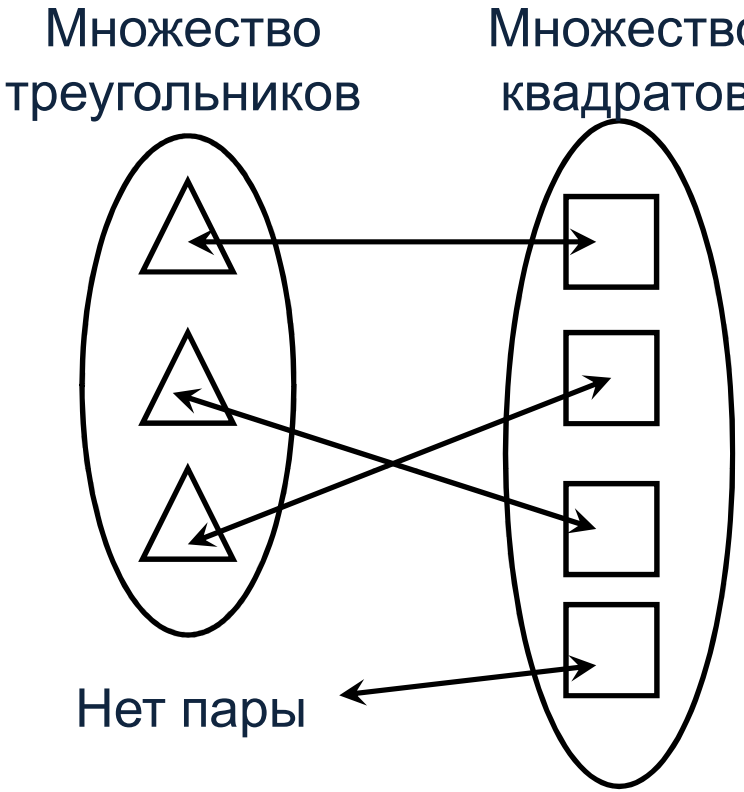
Добавим еще один квадрат.
Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная

Дискретная и непрерывная. В чем разница?

Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

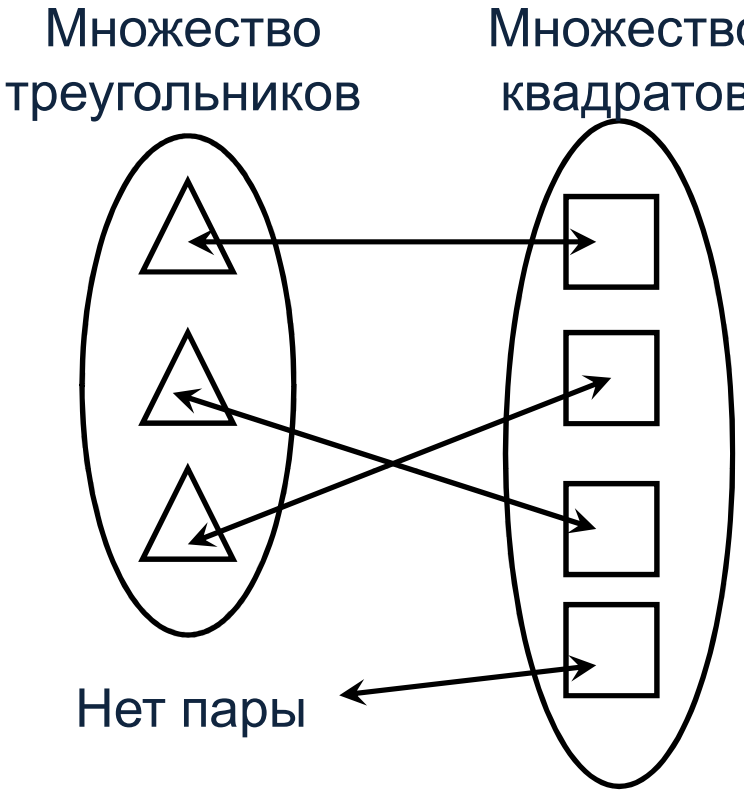
Следовательно, во множестве квадратов **больше** элементов

Непрерывная

Какой отрезок **длиннее**?

Дискретная и непрерывная. В чем разница?

Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

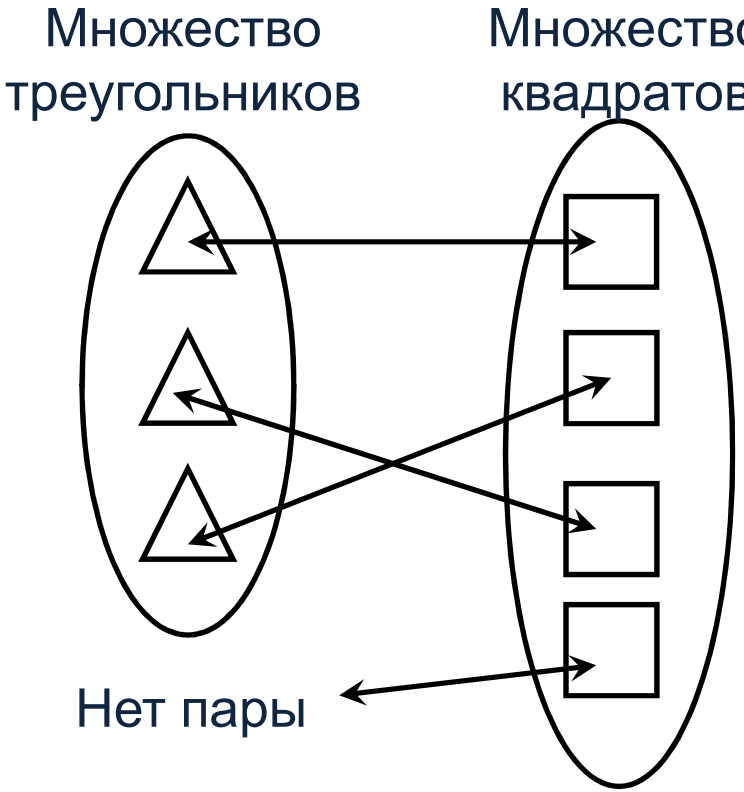
Следовательно, во множестве квадратов **больше** элементов

Непрерывная

Какой отрезок **длиннее**?
Нижний!

Дискретная и непрерывная. В чем разница?

Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

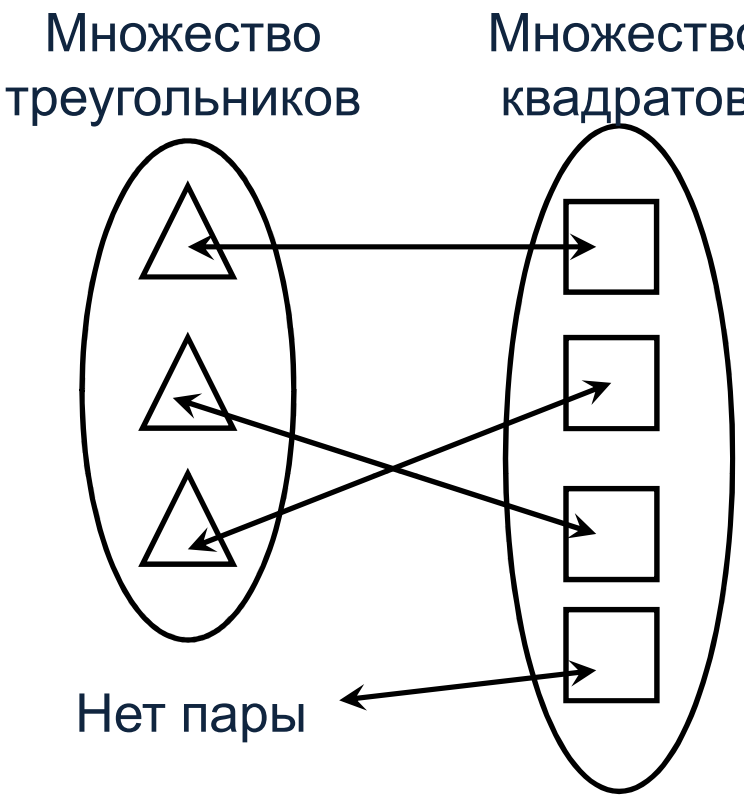
Непрерывная

Какой отрезок **длиннее**?
Нижний!

В каком отрезке **больше точек**?

Дискретная и непрерывная. В чем разница?

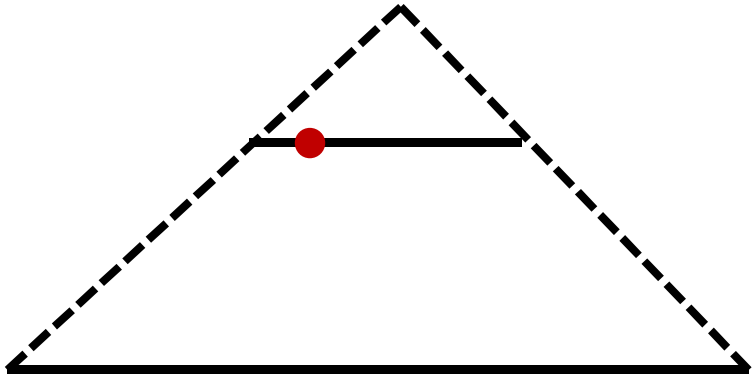
Дискретная



Добавим еще один квадрат. Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная



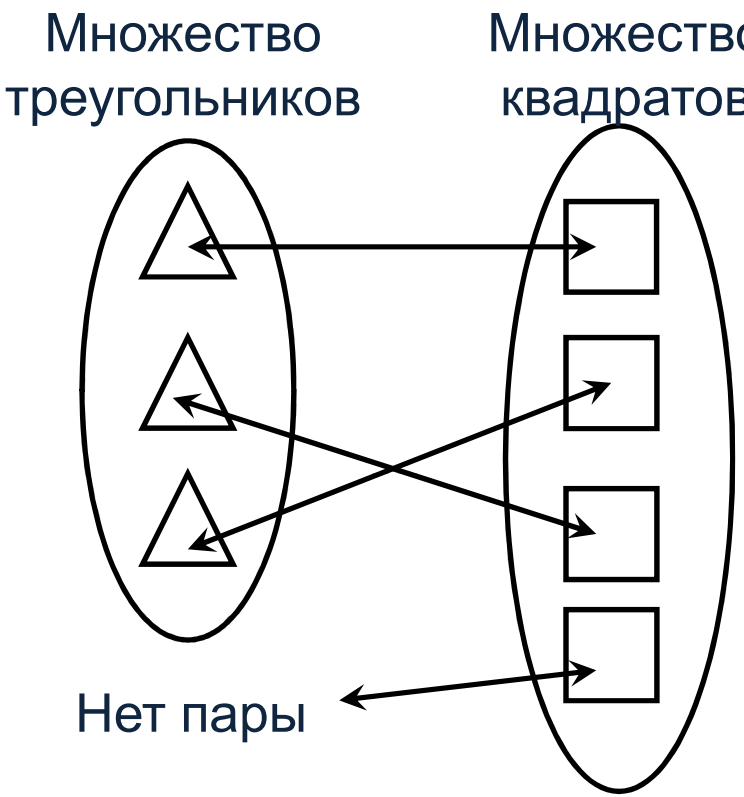
Какой отрезок **длиннее**?
Нижний!

В каком отрезке **больше точек**?
Одинаково!

Почему? **Установим взаимно-однозначное соответствие!**

Дискретная и непрерывная. В чем разница?

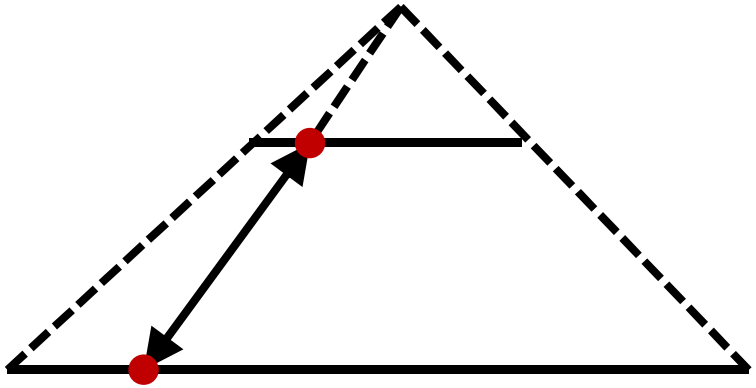
Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная



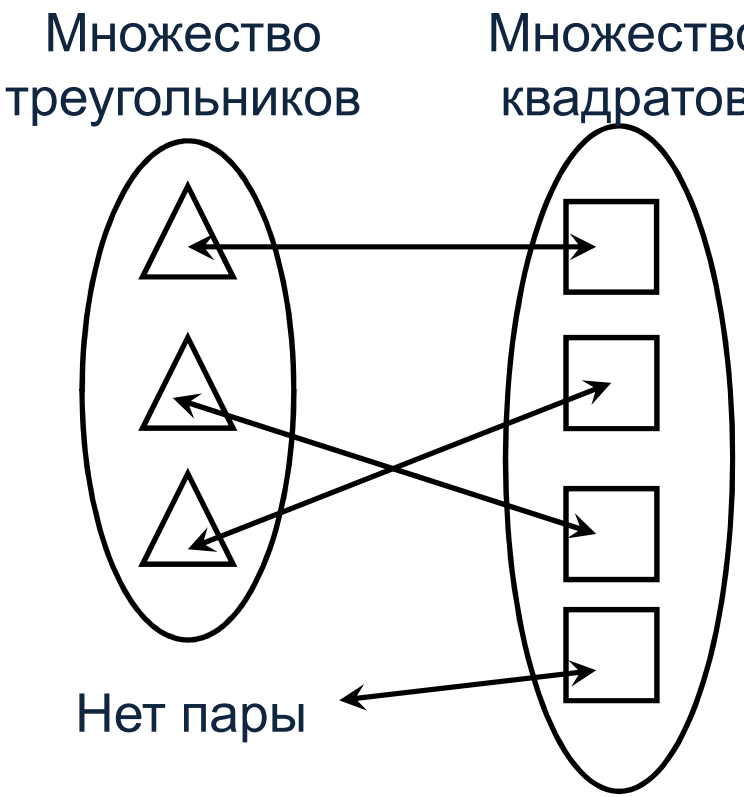
Какой отрезок **длиннее**?
Нижний!

В каком отрезке **больше точек**?
Одинаково!

Почему? **Установим взаимно-однозначное соответствие!**

Дискретная и непрерывная. В чем разница?

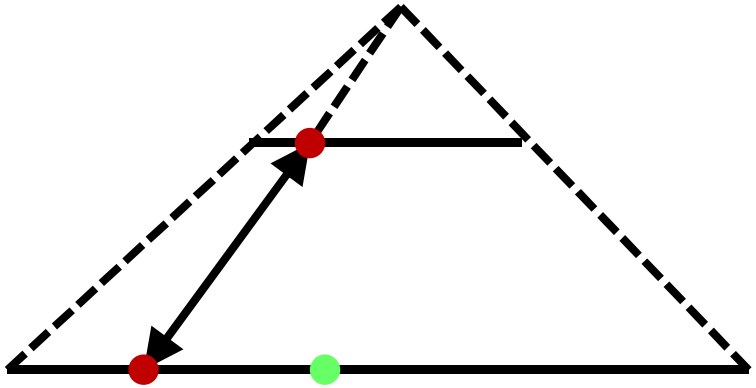
Дискретная



Добавим еще один квадрат. Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная



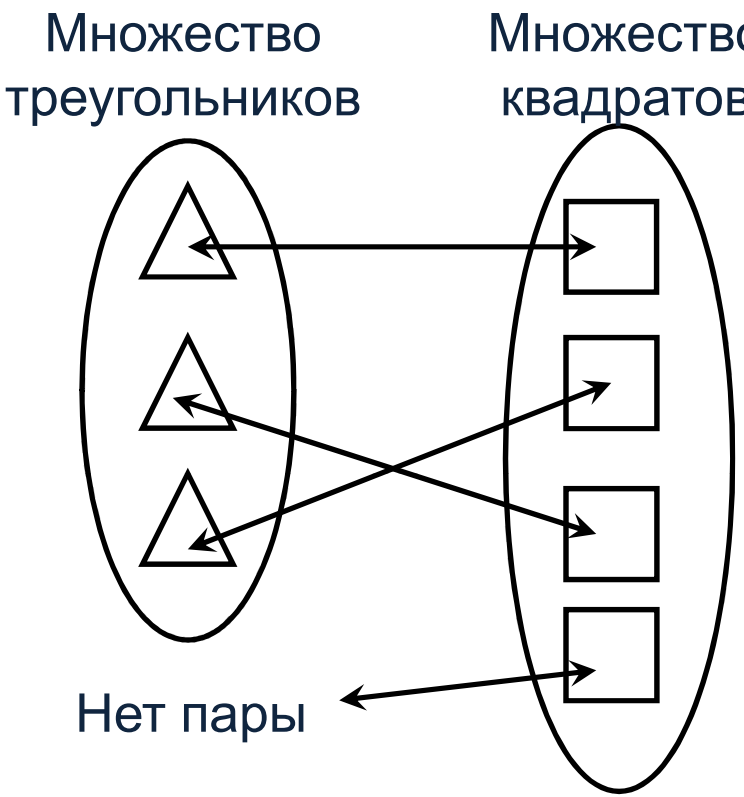
Какой отрезок **длиннее**?
Нижний!

В каком отрезке **больше точек**?
Одинаково!

Почему? **Установим взаимно-однозначное соответствие!**

Дискретная и непрерывная. В чем разница?

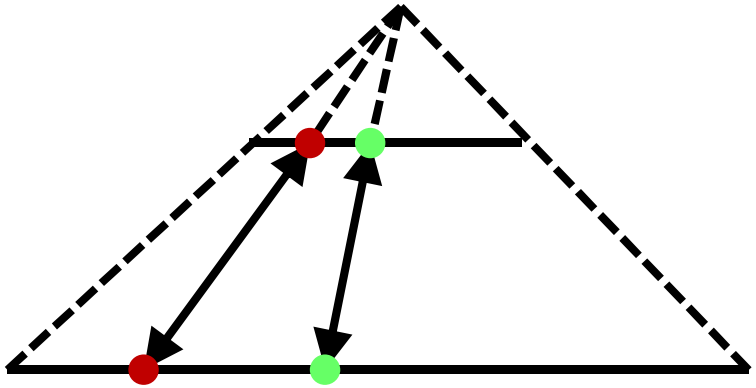
Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная



Какой отрезок **длиннее**?
Нижний!

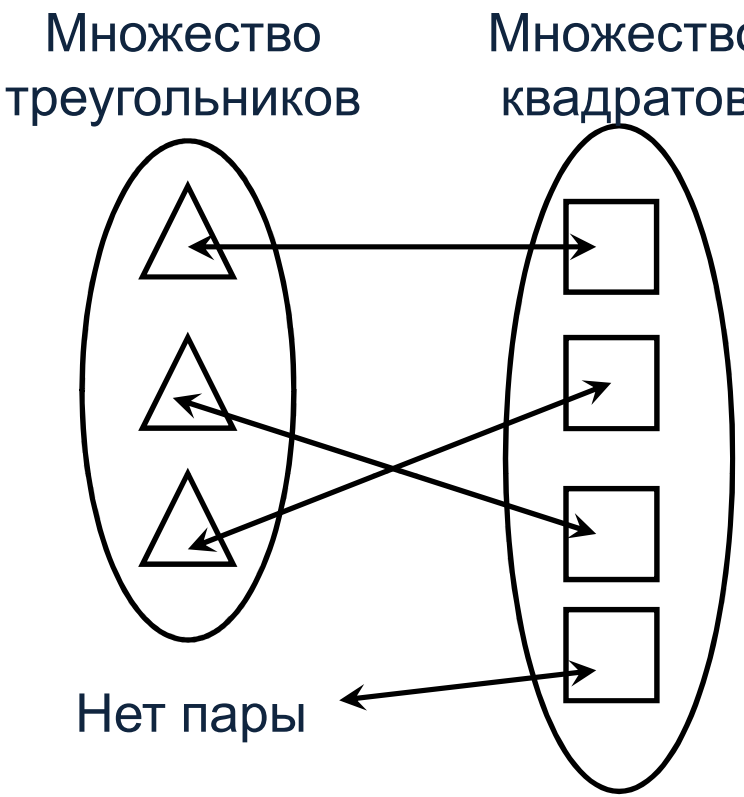
В каком отрезке **больше точек**?
Одинаково!

Почему? **Установим взаимно-однозначное соответствие!**

Разным точкам внизу – разные вверху, и наоборот.
Все точки разбились на пары.

Дискретная и непрерывная. В чем разница?

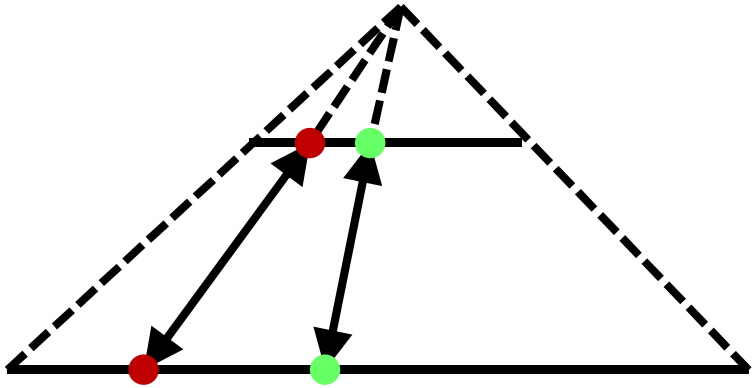
Дискретная



Добавим еще один квадрат.
Тогда, ему не найдется пары.

Следовательно, во множестве квадратов **больше** элементов

Непрерывная



Какой отрезок **длиннее**?
Нижний!

В каком отрезке **больше точек**?
Одинаково!

Почему? **Установим взаимно-однозначное соответствие!**

Разным точкам внизу – разные вверху, и наоборот.
Все точки разбились на пары.

Бесконечность – хитрая штука…

МГУ – ШКОЛЕ

Шаг в бесконечность...

Шаг в бесконечность...

МГУ – ШКОЛЕ

Хороший способ спорить и выигрывать.

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:
$0{,}9 < 1$

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:

$0{,}9 < 1$

$0{,}99 < 1$

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:

$0{,}9 < 1$

$0{,}99 < 1$

$0{,}999 < 1$

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:

$0{,}9 < 1$

$0{,}99 < 1$

$0{,}999 < 1$

$0{,}999\ldots < 1$

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:

$0{,}9 < 1$

$0{,}99 < 1$

$0{,}999 < 1$

$0{,}999\ldots < 1$

А какие еще могут быть варианты: $0{,}999\ldots$? 1

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:
0,9 < 1
0,99 < 1
0,999 < 1
0,999… < 1

А какие еще могут быть варианты: 0,999… ? 1
0,999… = 1
0,999… и 1 несравнимы

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:
0,9 < 1
0,99 < 1
0,999 < 1
0,999… < 1

А какие еще могут быть варианты: 0,999… ? 1
0,999… = 1
0,999… и 1 несравнимы
А спорим, что равно!?

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:
0,9 < 1
0,99 < 1
0,999 < 1
0,999… < 1

А какие еще могут быть варианты: 0,999… ? 1
0,999… = 1
0,999… и 1 несравнимы
А спорим, что равно!?

x = 0,999…
10x = 9,999…
9x = 10x – x = 9,999… – 0,999... = 9
Получаем уравнение 9x = 9

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

Сначала затравка… Верно ли, что:
0,9 < 1
0,99 < 1
0,999 < 1
0,999… < 1

А какие еще могут быть варианты: 0,999… ? 1
0,999… = 1
0,999… и 1 несравнимы
А спорим, что равно!?

x = 0,999…
10x = 9,999…
9x = 10x – x = 9,999… – 0,999... = 9
Получаем уравнение 9x = 9
x = 1, т.е. 0,999… = 1

МГУ – ШКОЛЕ

Шаг в бесконечность...

Хороший способ спорить и выигрывать.

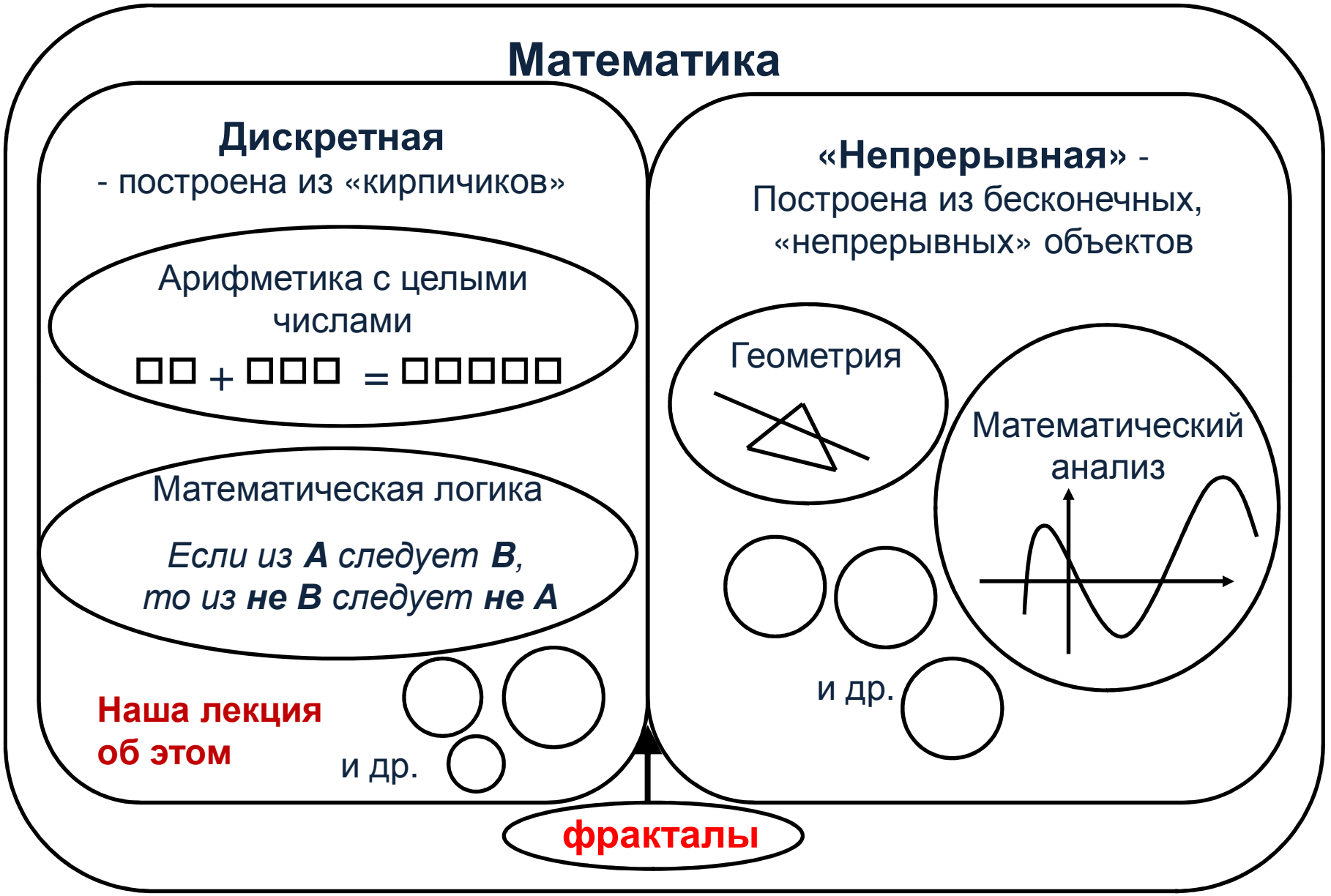
Сначала затравка… Верно ли, что:
0,9 < 1
0,99 < 1
0,999 < 1
0,999… < 1

А какие еще могут быть варианты: 0,999… ? 1
0,999… = 1
0,999… и 1 несравнимы
А спорим, что равно!?

x = 0,999…
10x = 9,999…
9x = 10x – x = 9,999… – 0,999... = 9
Получаем уравнение 9x = 9
x = 1, т.е. 0,999… = 1

А это нормально: 6/2 = 3 ?

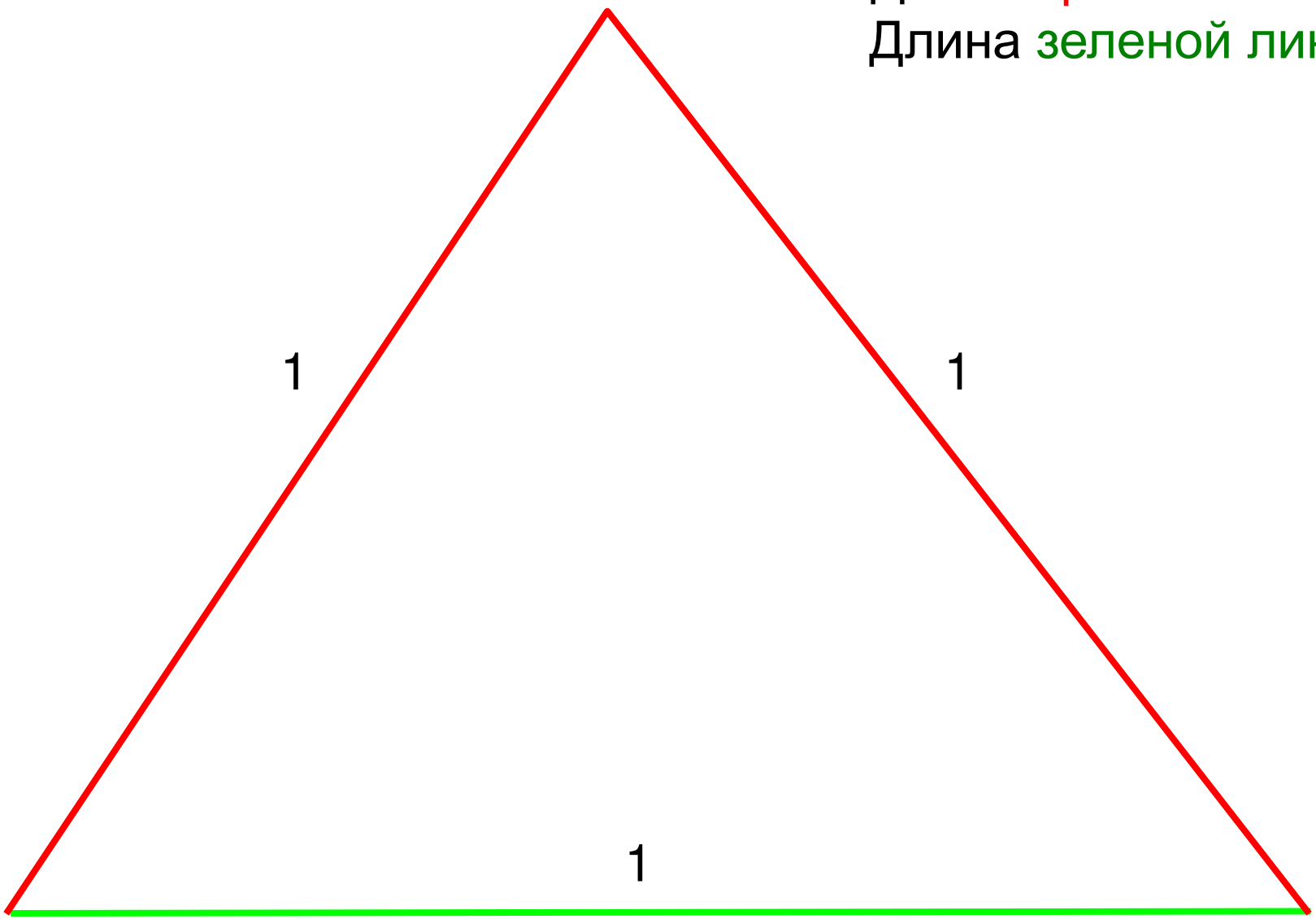
Фракталы


Математика
Дискретная
- построена из «кирпичиков»
Арифметика с целыми числами
□□ + □□□ = □□□□□
Математическая логика
Если из А следует В,
то из не В следует не А
Наша лекция об этом
и др.
«Непрерывная» -
Построена из бесконечных, «непрерывных» объектов
Геометрия
Математический анализ
и др.
фракталы

Фракталы – странная кривая.

Длина красной линии = 2

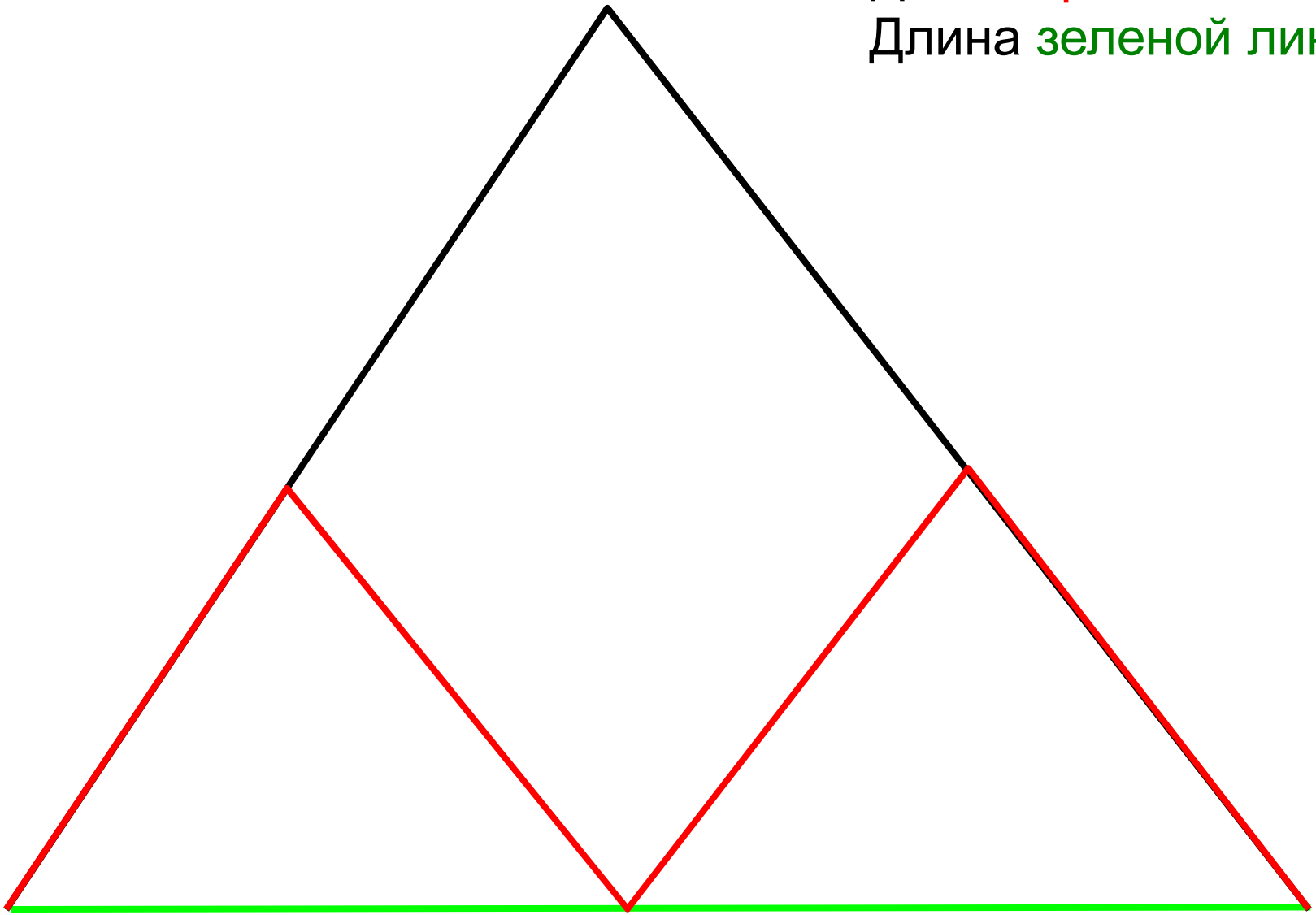
Длина зеленой линии = 1



Фракталы – странная кривая.

Длина красной линии = 2

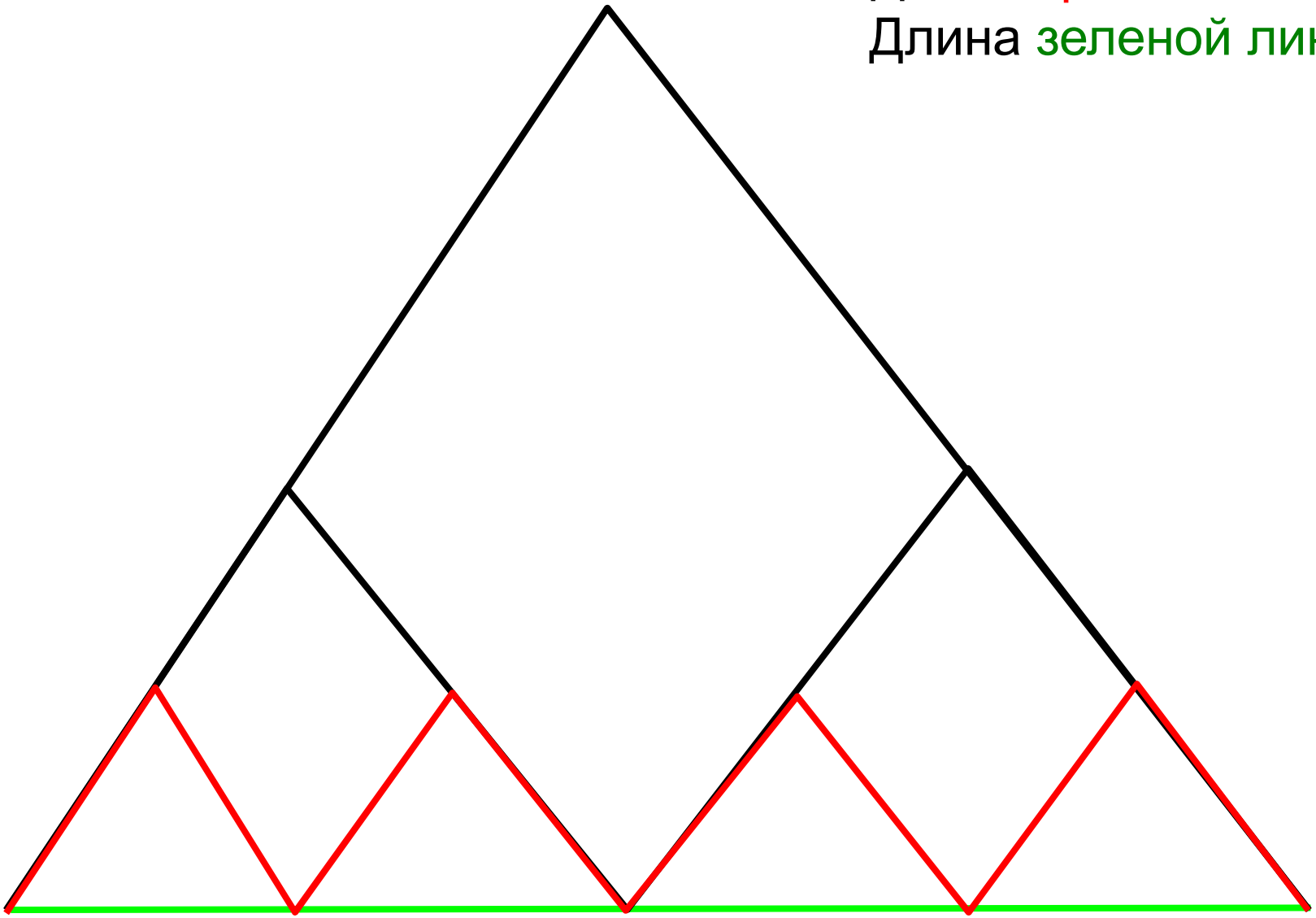
Длина зеленой линии = 1



Фракталы – странная кривая.

Длина красной линии = 2

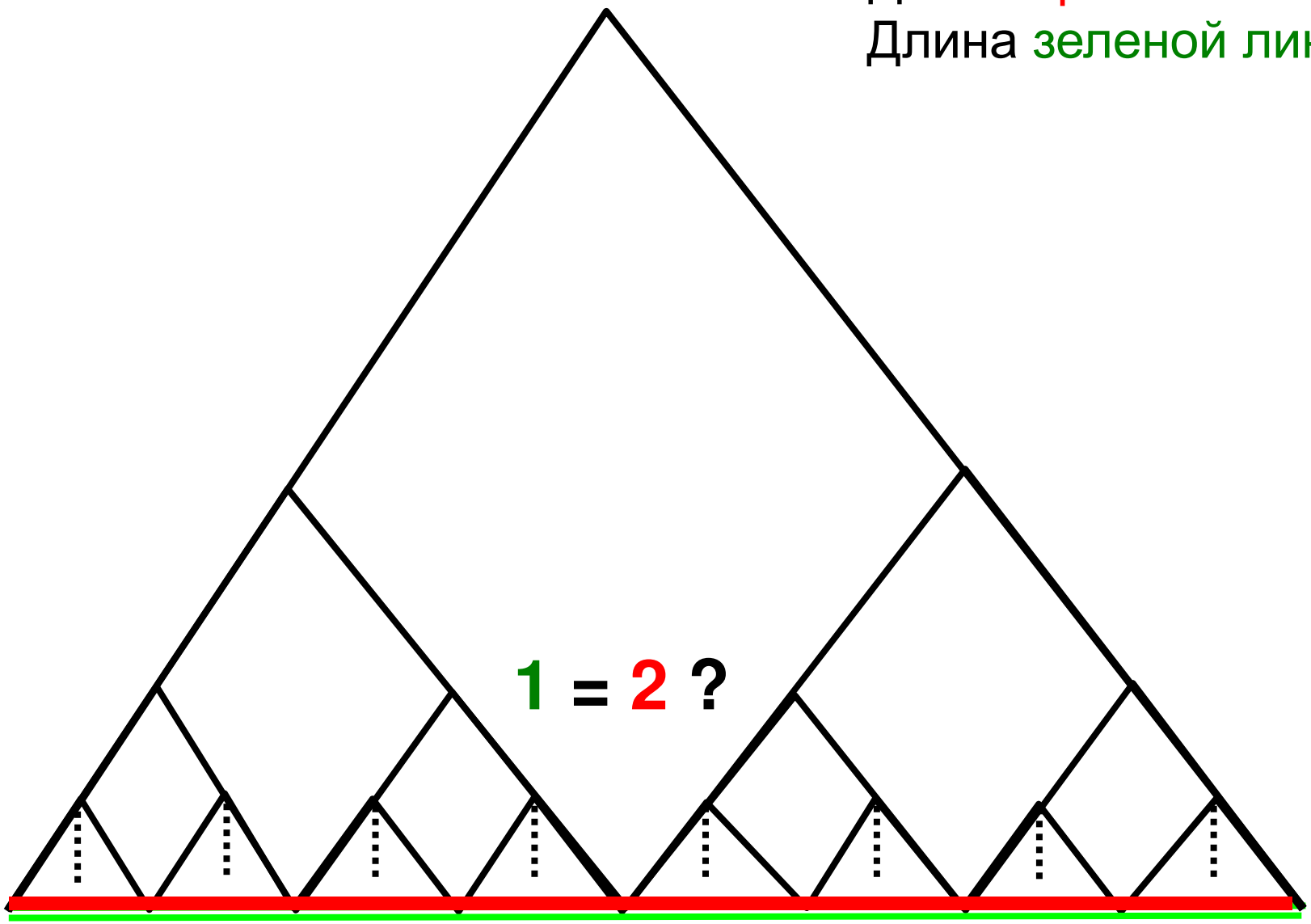
Длина зеленой линии = 1



Фракталы – странная кривая.

Длина красной линии = 2

Длина зеленой линии = 1



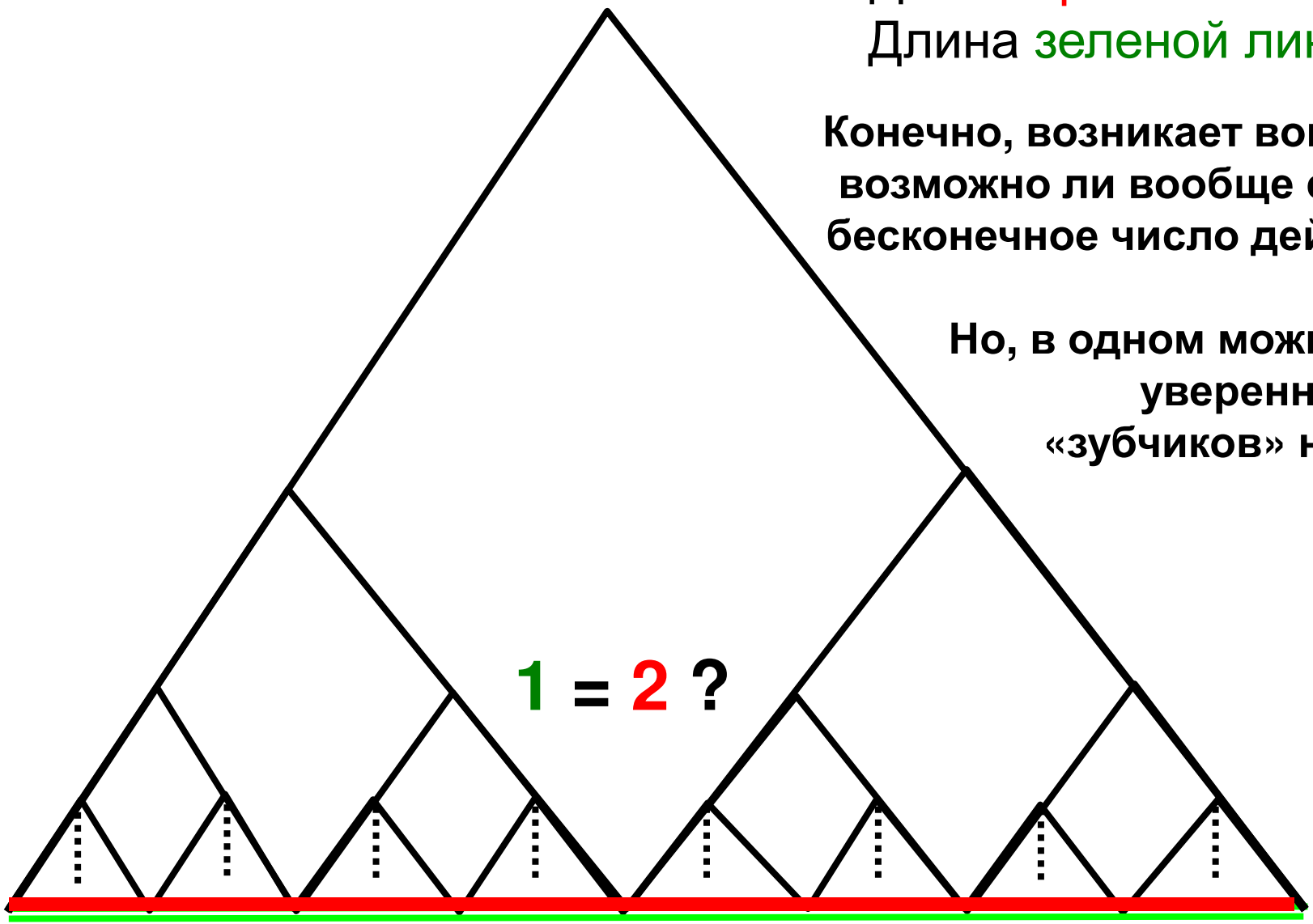
Фракталы – странная кривая.

Длина красной линии = 2

Длина зеленой линии = 1

Конечно, возникает вопрос – а возможно ли вообще сделать бесконечное число действий?

Но, в одном можно быть уверенным, что «зубчиков» не будет



Фракталы – странная кривая.

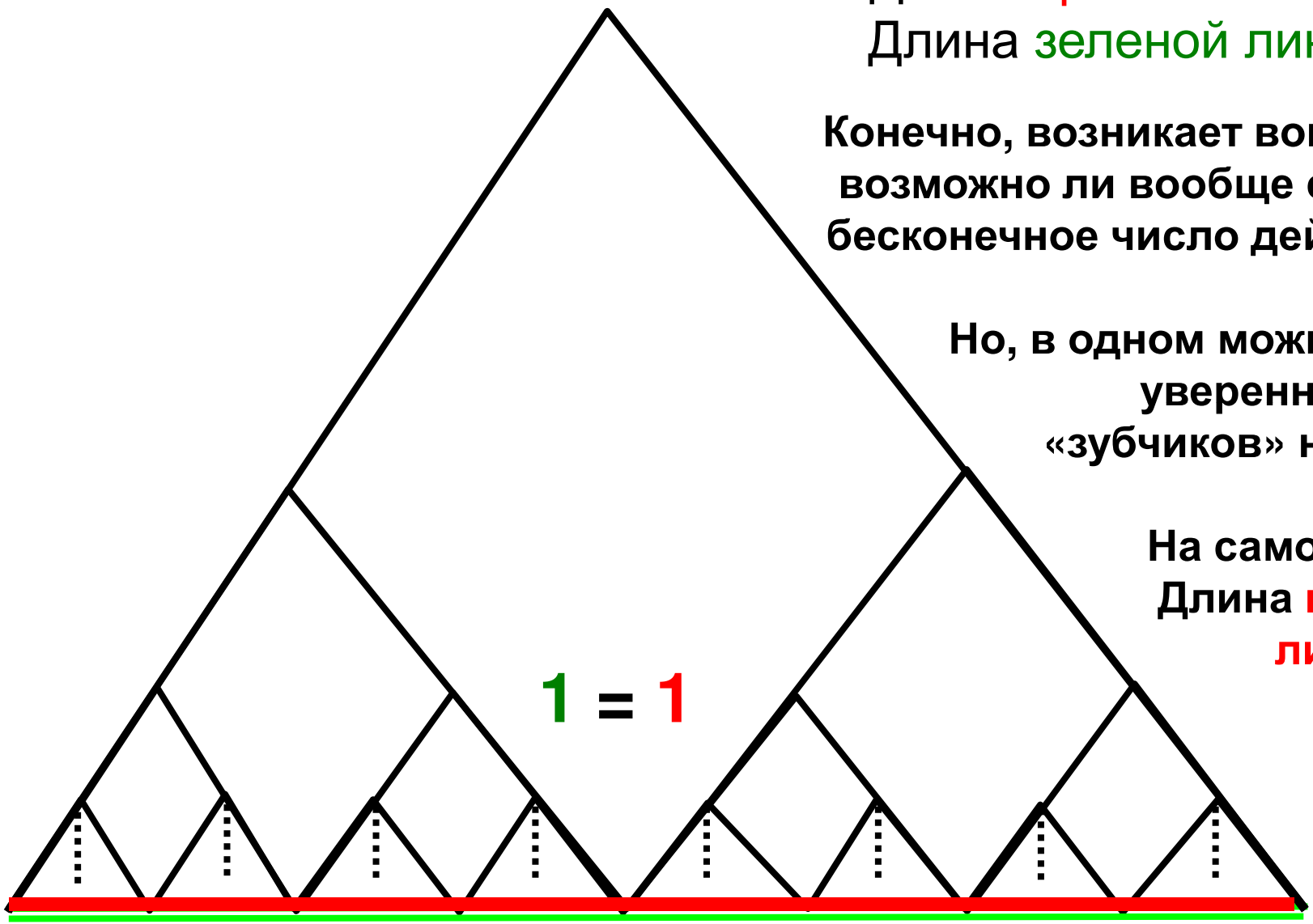
Длина красной линии = 2

Длина зеленой линии = 1

Конечно, возникает вопрос – а возможно ли вообще сделать бесконечное число действий?

Но, в одном можно быть уверенным, что «зубчиков» не будет

На самом деле:
Длина красной линии = 1



Фракталы – странная кривая.

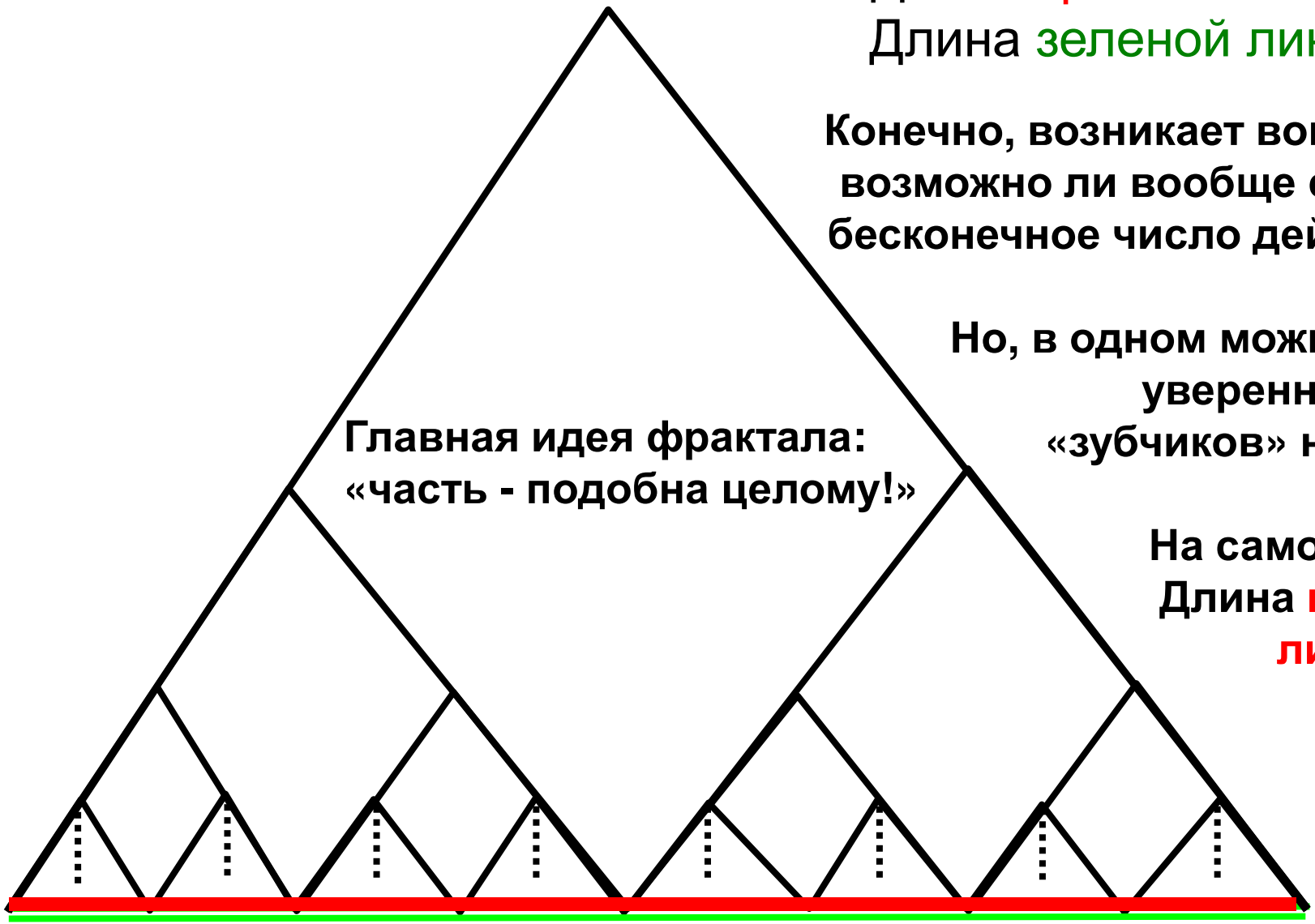
Длина красной линии = 2

Длина зеленой линии = 1

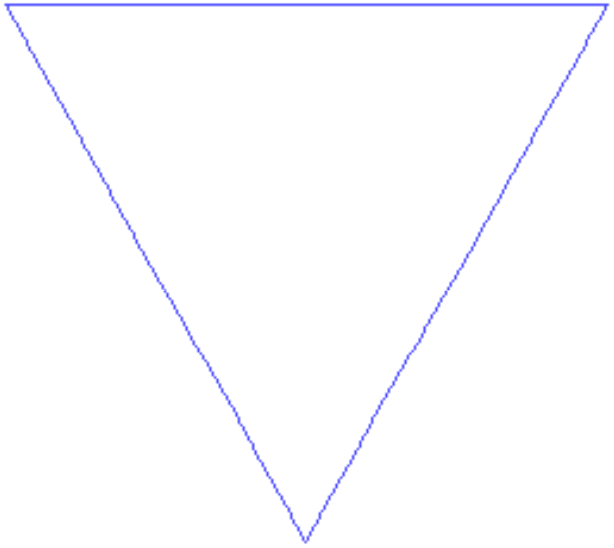
Конечно, возникает вопрос – а возможно ли вообще сделать бесконечное число действий?

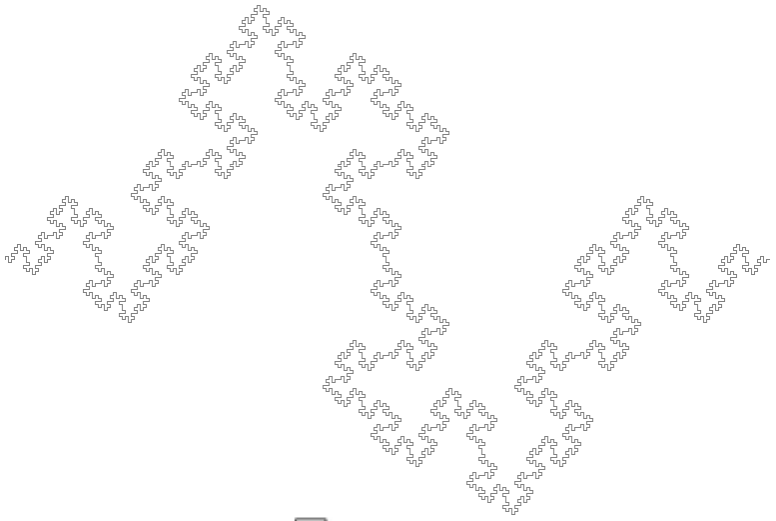
Но, в одном можно быть уверенным, что «зубчиков» не будет

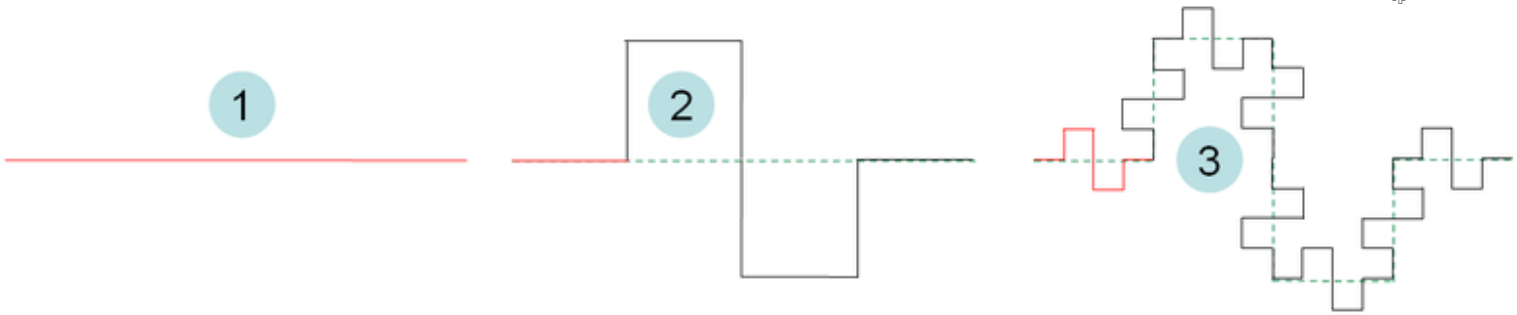
На самом деле: Длина красной линии = 1



Фракталы – кривая Коха

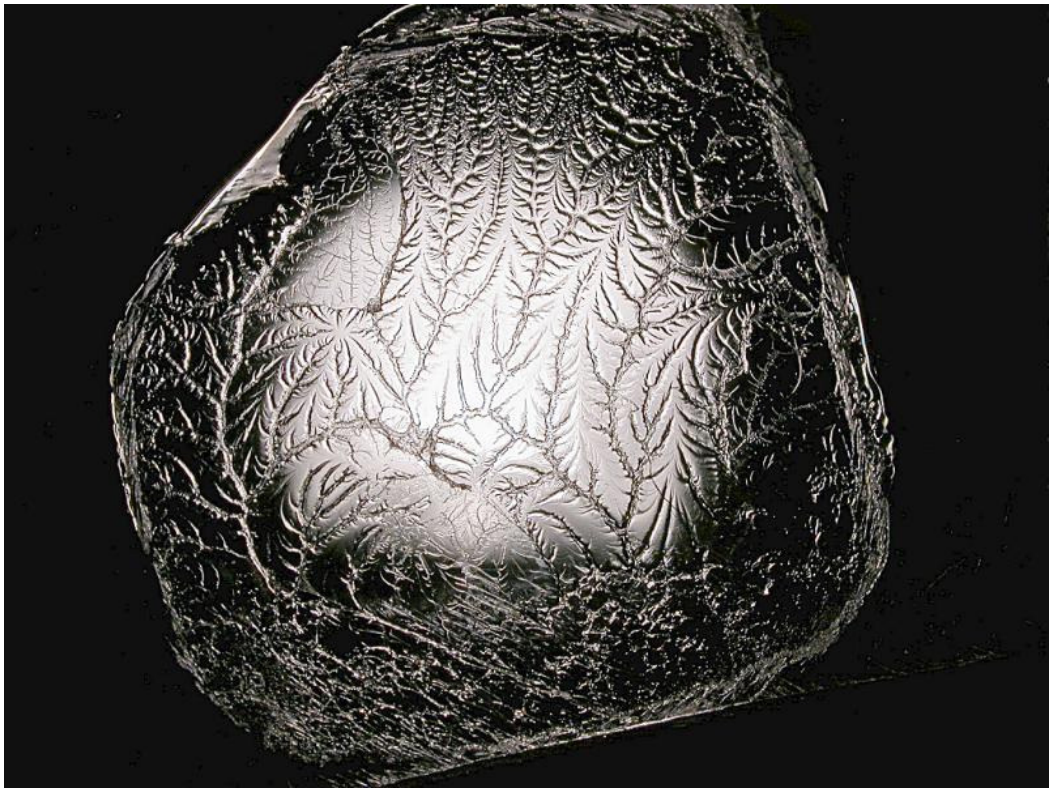






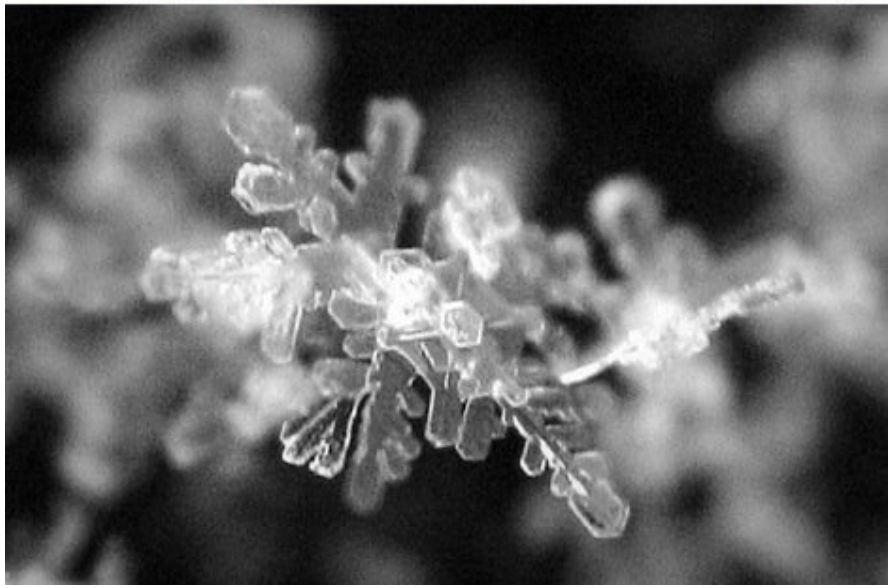
МГУ – ШКОЛЕ

Фракталы в природе



МГУ – ШКОЛЕ

Фракталы в природе

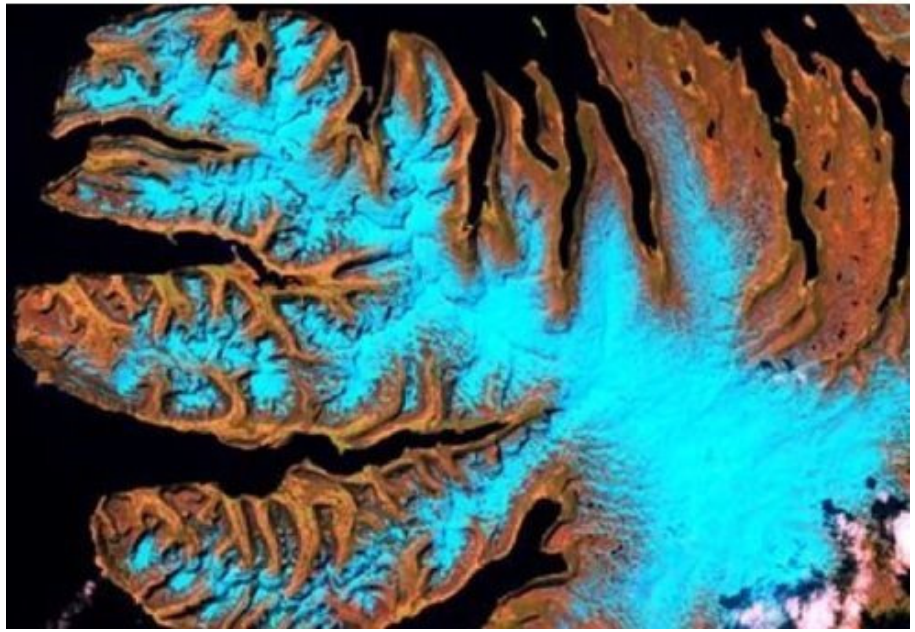


МГУ – ШКОЛЕ

Фракталы в природе

МГУ – ШКОЛЕ

Фракталы в природе



МГУ – ШКОЛЕ

Фракталы в природе

МГУ – ШКОЛЕ

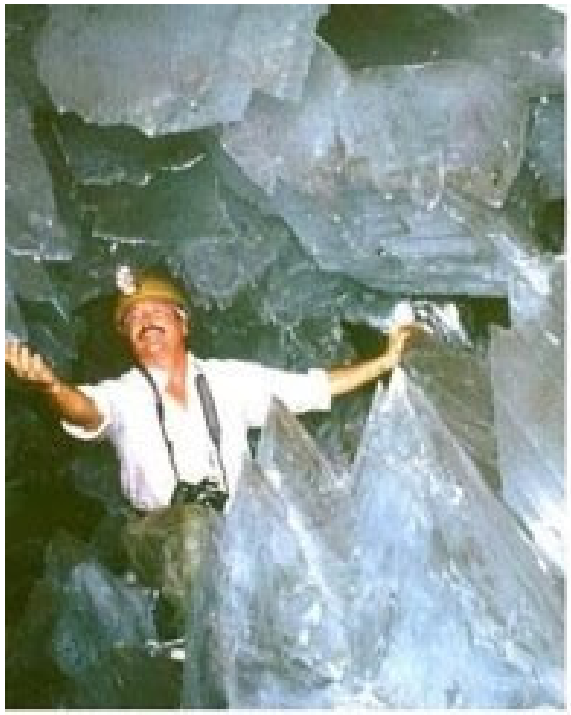
Фракталы в природе

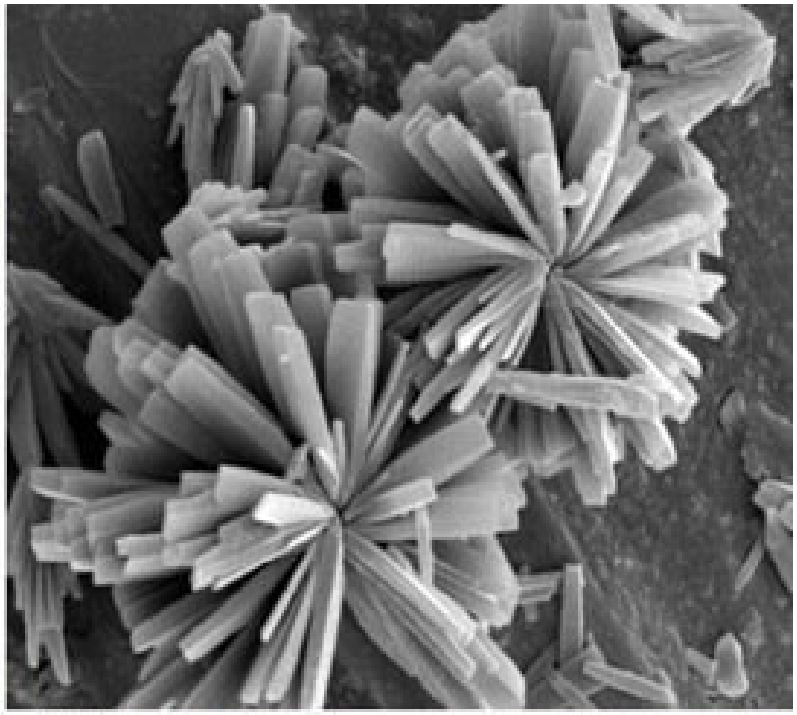
МГУ – ШКОЛЕ

Фракталы в природе

МГУ – ШКОЛЕ

Фракталы в природе



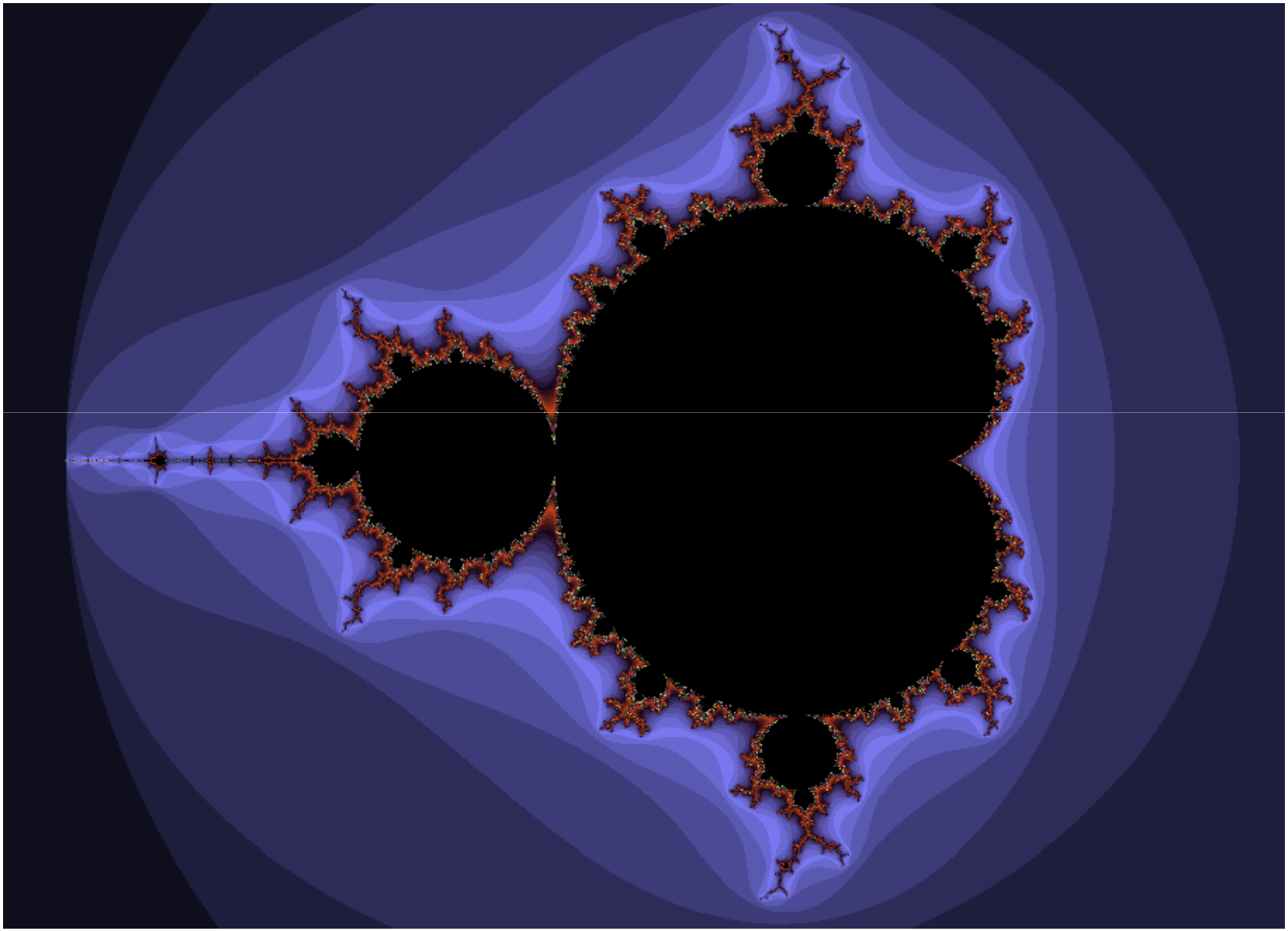


МГУ – ШКОЛЕ

Фракталы в природе

МГУ – ШКОЛЕ

Фракталы на компьютере



4-х мерный куб

Видели ли Вы когда-нибудь 4-х мерный куб?

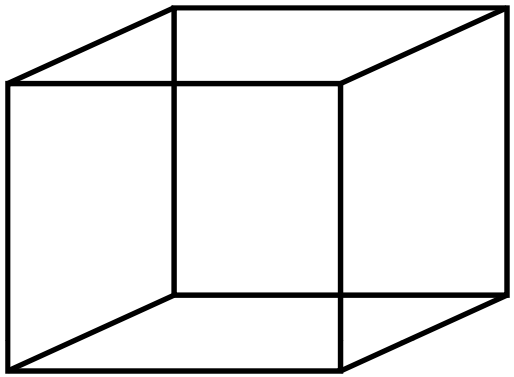
А держали ли Вы когда-нибудь 4-х мерный куб в руках?

3-х мерный куб

Но начнем с 3-х мерного куба…

3-х мерный куб

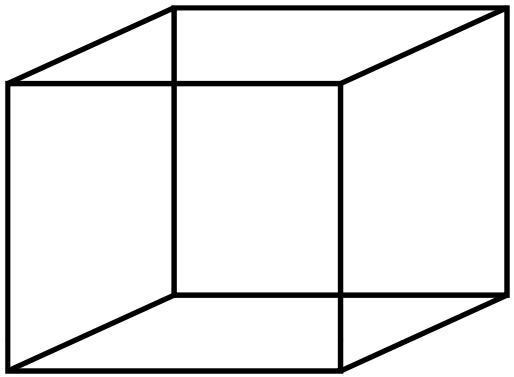
Но начнем с 3-х мерного куба…



Что вы видите?

3-х мерный куб

Но начнем с 3-х мерного куба…

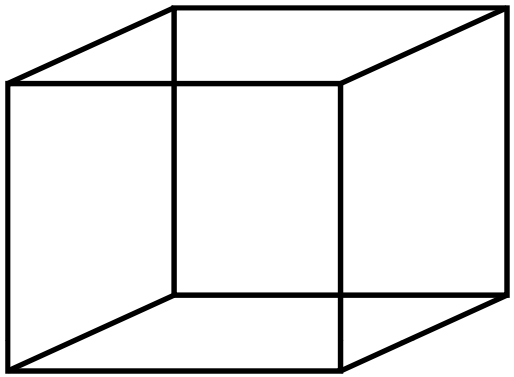


Что вы видите?

3-х мерный куб?

3-х мерный куб

Но начнем с 3-х мерного куба…



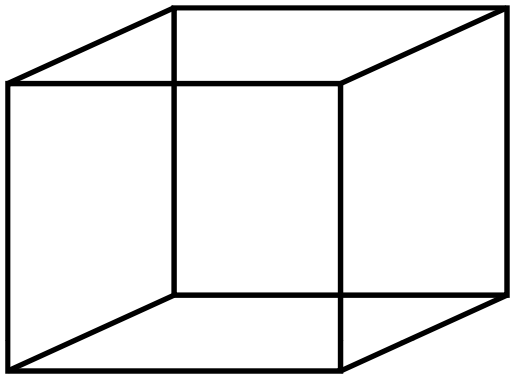
Что вы видите?

3-х мерный куб?

Вы видите 3-х мерный куб в 2-х мерном пространстве!

3-х мерный куб

Но начнем с 3-х мерного куба…



Что вы видите?

3-х мерный куб?

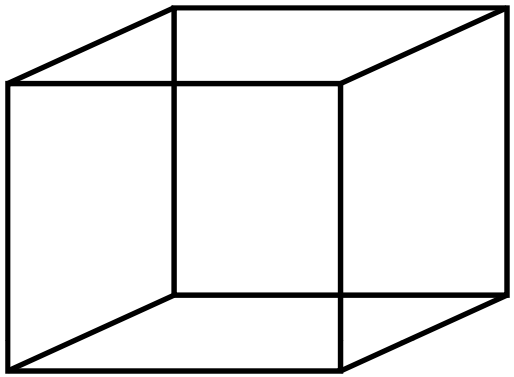
Вы видите 3-х мерный куб в 2-х мерном пространстве!

Следовательно, можно увидеть объект из пространства с большей размерностью в пространстве с меньшей размерностью.

Объект можно «вжать» (спроецировать) в пространство меньшей размерности.

3-х мерный куб

Но начнем с 3-х мерного куба…



Что вы видите?

3-х мерный куб?

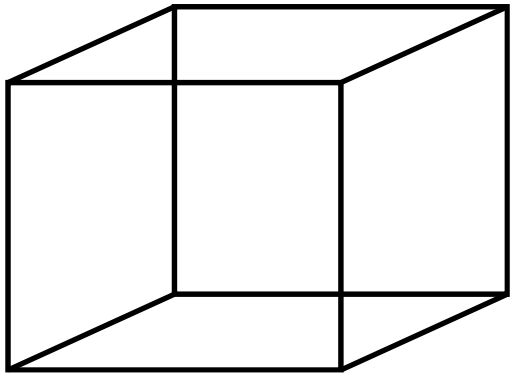
Вы видите 3-х мерный куб в 2-х мерном пространстве!

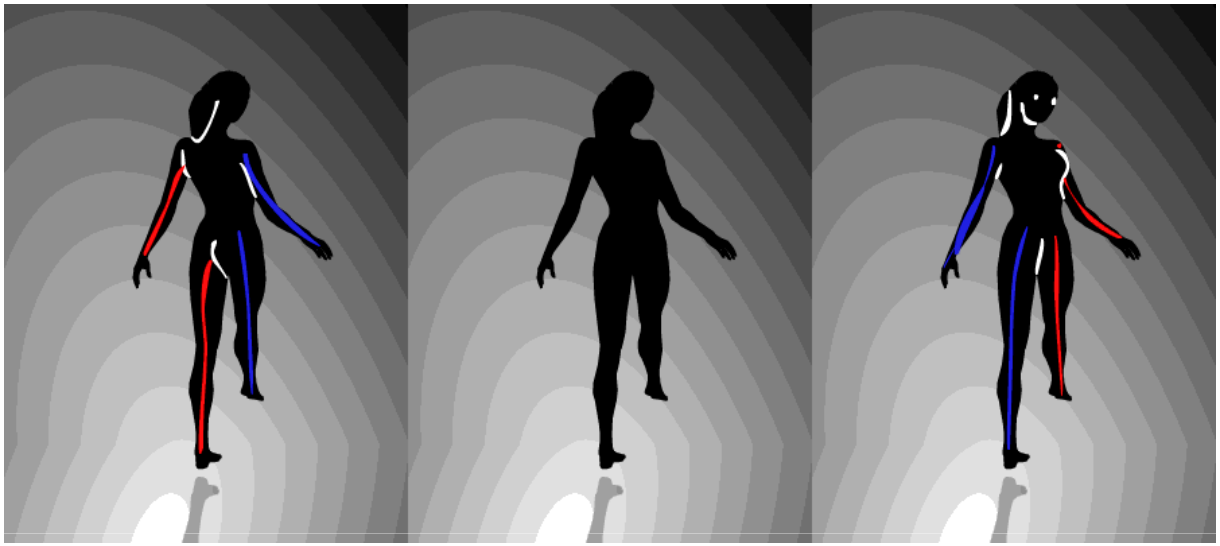
Следовательно, можно увидеть объект из пространства с большей размерностью в пространстве с меньшей размерностью.

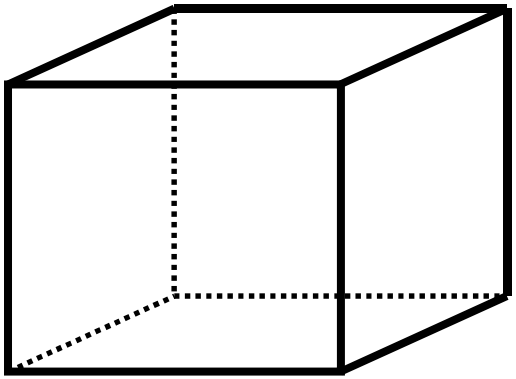
Объект можно «вжать» (спроецировать) в пространство меньшей размерности.

А мы на него смотрим сверху или снизу?

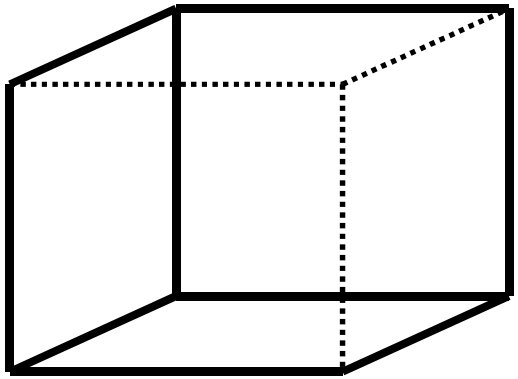
3-х мерный куб







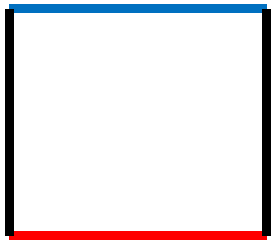
А мы на него смотрим сверху или снизу?



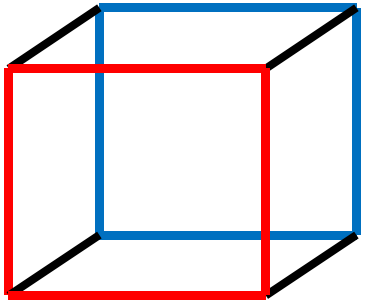
4-х мерный куб

Размерность 1

Размерность 2



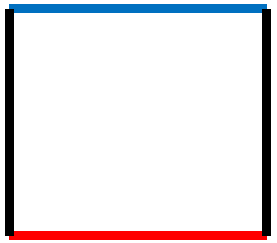
Размерность 3



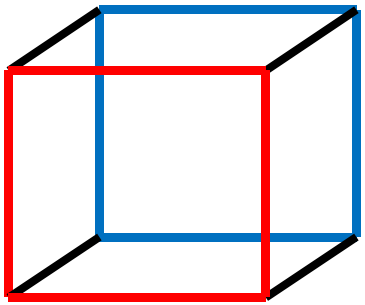
4-х мерный куб

Размерность 1

Размерность 2



Размерность 3



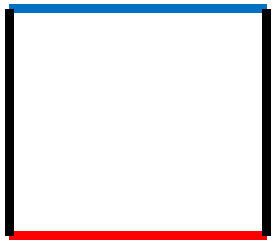
Технология прыжка в следующую размерность:

Берем две копии из предыдущей размерности и соединяем параллельными отрезками

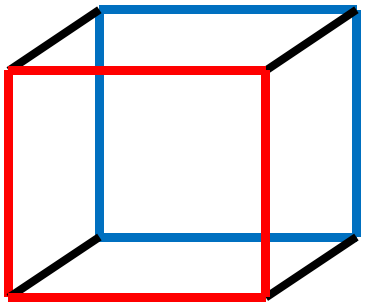
4-х мерный куб

Размерность 1

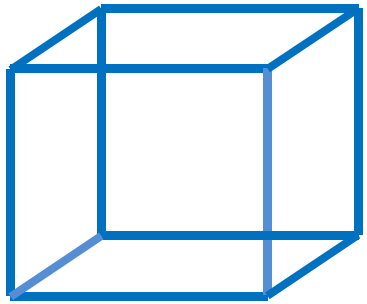
Размерность 2

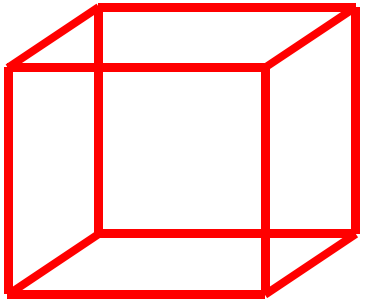


Размерность 3



Размерность 4





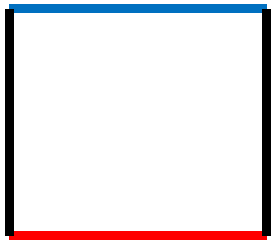
Технология прыжка в следующую размерность:

Берем две копии из предыдущей размерности и соединяем параллельными отрезками

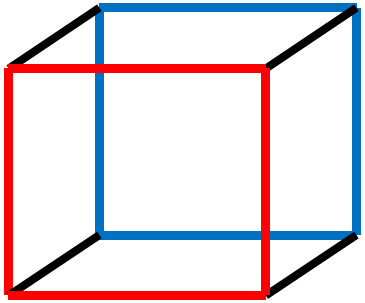
4-х мерный куб

Размерность 1

Размерность 2



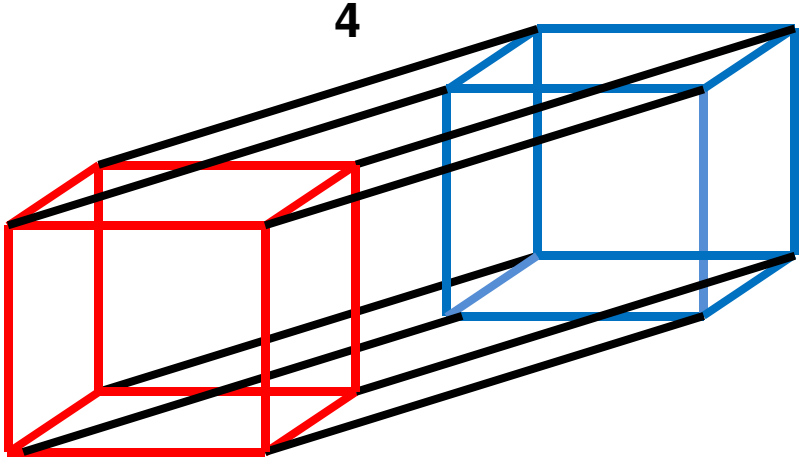
Размерность 3



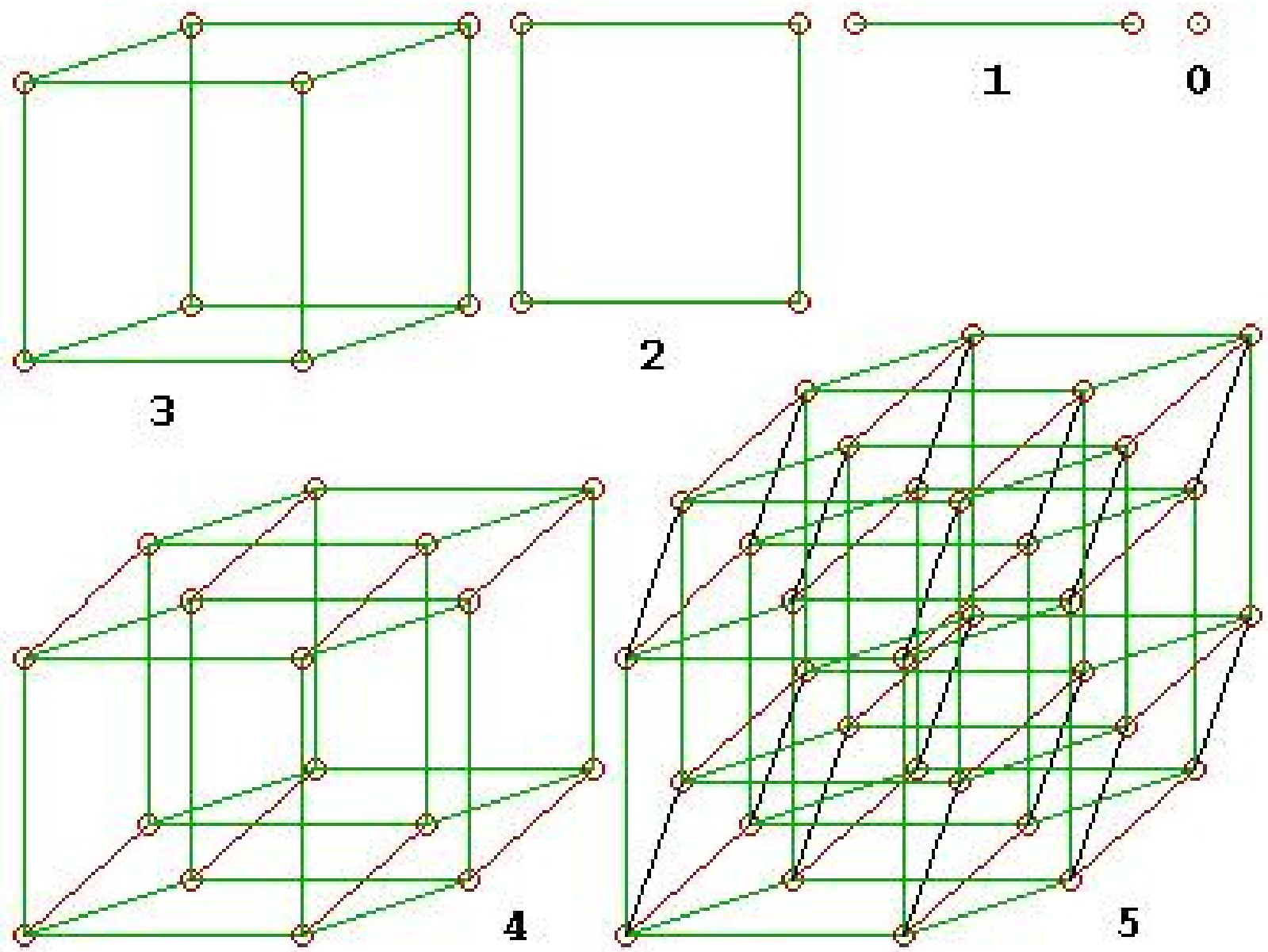
Технология прыжка в следующую размерность:

Берем две копии из предыдущей размерности и соединяем параллельными отрезками

Размерность 4



5-и мерный куб



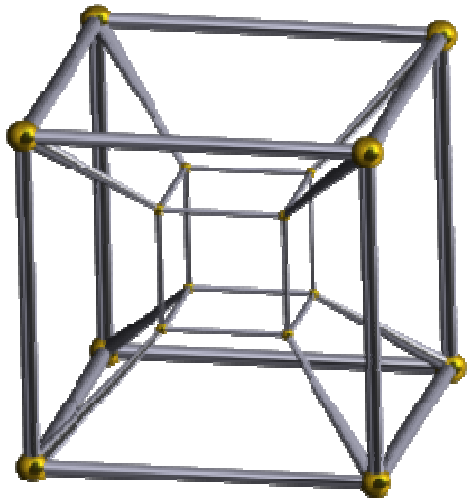
МГУ – ШКОЛЕ

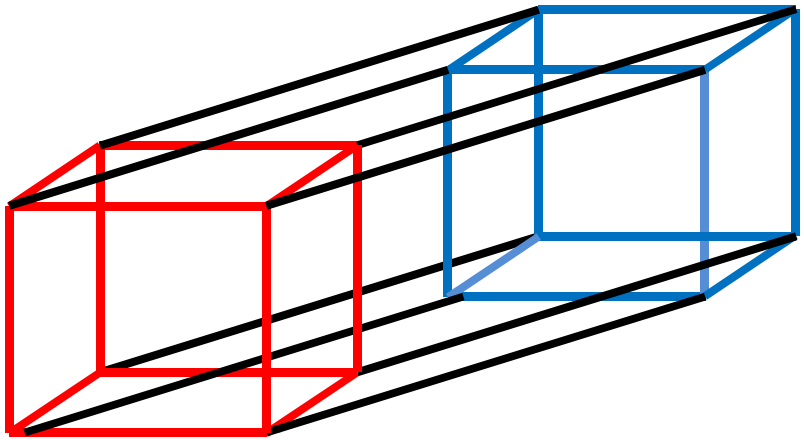
4-х мерный куб?

Иногда можно встретить небольшую ошибку в изображении 4-х мерного куба

4-х мерный куб?

Иногда можно встретить небольшую ошибку в изображении 4-х мерного куба

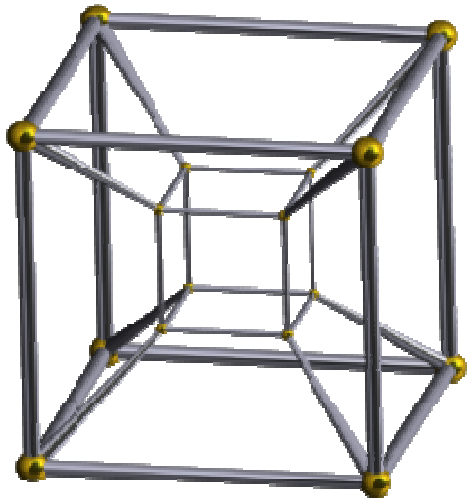


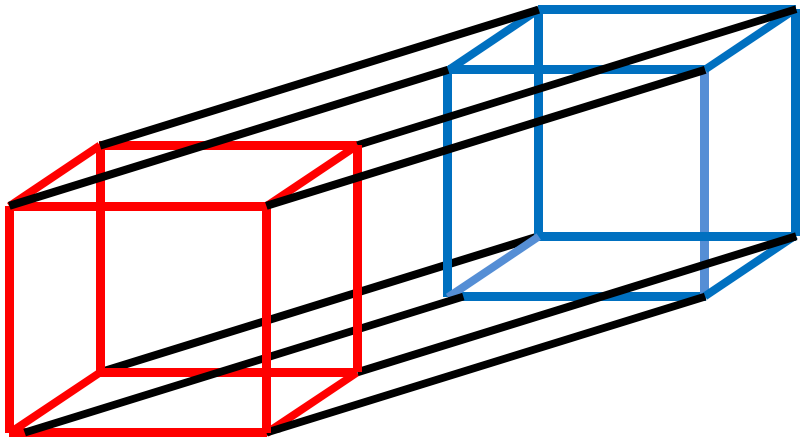


В чем ошибка этого изображения?
Казалось бы все тоже самое…

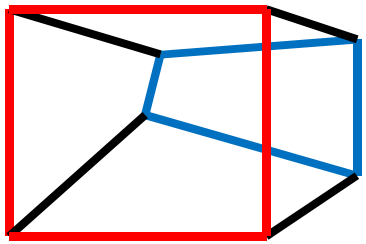
4-х мерный куб?

Иногда можно встретить небольшую ошибку в изображении 4-х мерного куба



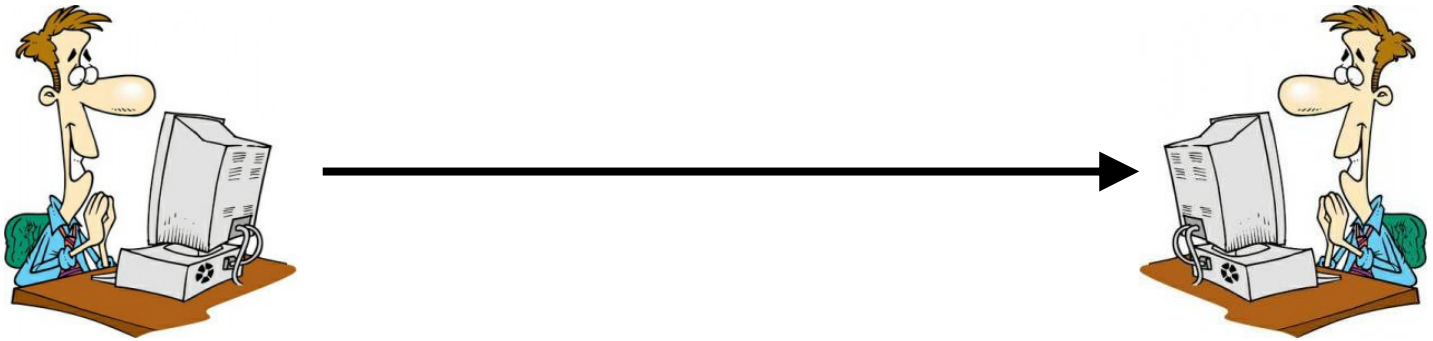


В чем ошибка этого изображения?
Казалось бы все тоже самое…
Но соответствующие стороны не параллельны!
А при проекции, параллельные прямые переходят в параллельные



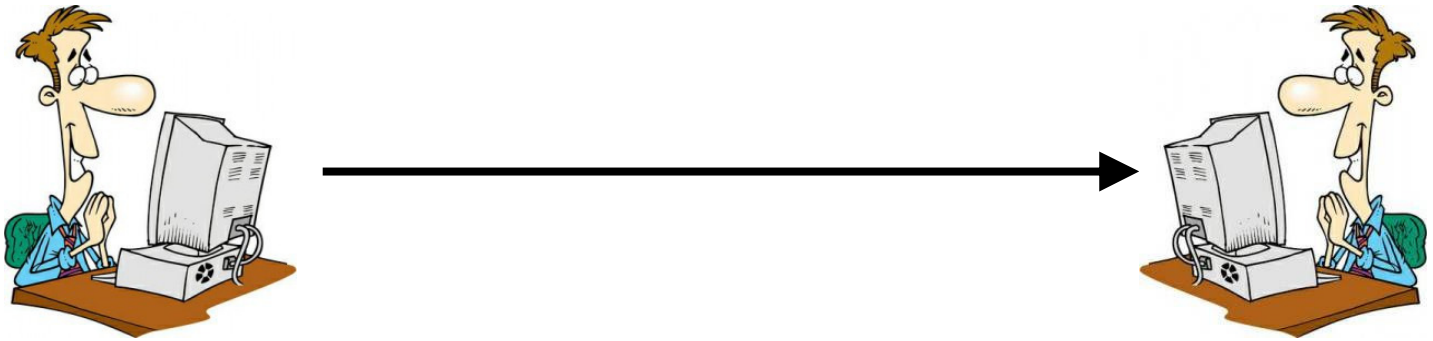
Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.



Кодирование информации и защита от помех

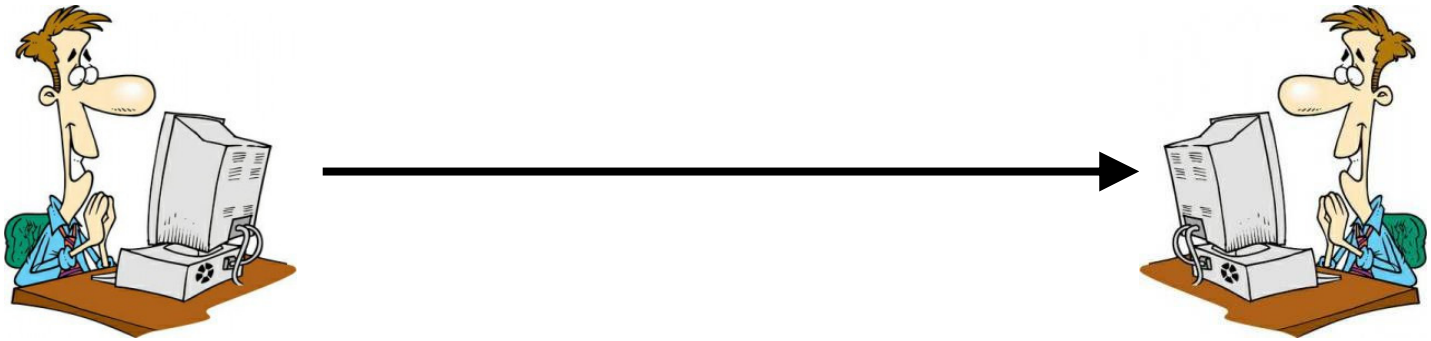
Задача: Надо отправить «да» или «нет» от одного пользователя к другому.



Информация по каналу связи передается в битах.
Бит – одна ячейка информации, содержащая 0 или 1

Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.

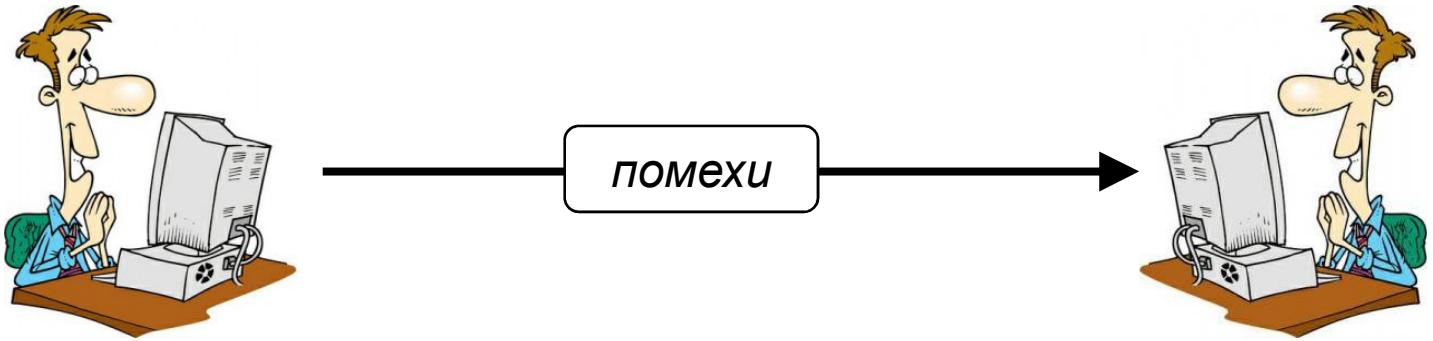


Информация по каналу связи передается в битах.
Бит – одна ячейка информации, содержащая 0 или 1

Если нет помех в канале связи (произвольных изменений 0 на 1 и обратно), то можно:
Если «Да», то передать 1
Если «Нет», то передать 0

Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.

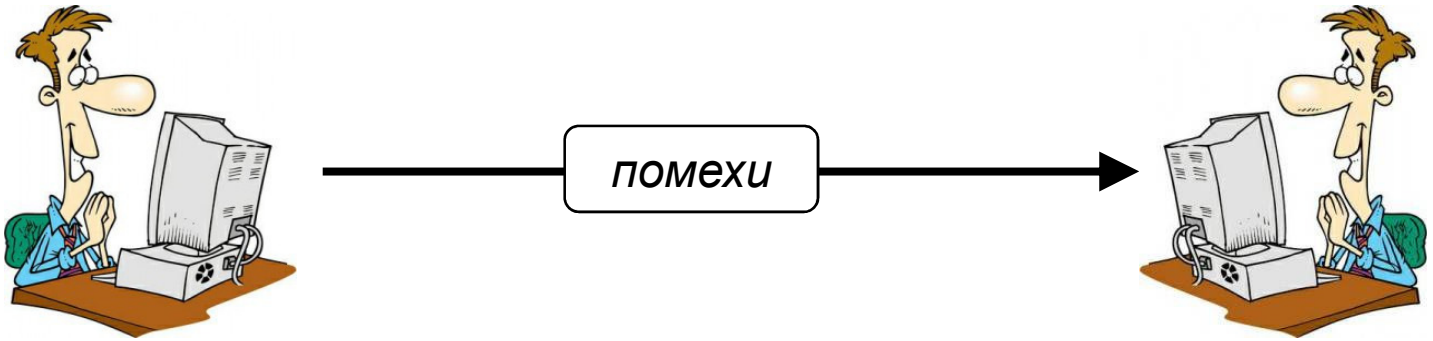


Информация по каналу связи передается в битах.
Бит – одна ячейка информации, содержащая 0 или 1

Путь есть помехи в канале связи и известна, например, вероятность их происхождения.

Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.



Информация по каналу связи передается в битах.
Бит – одна ячейка информации, содержащая 0 или 1

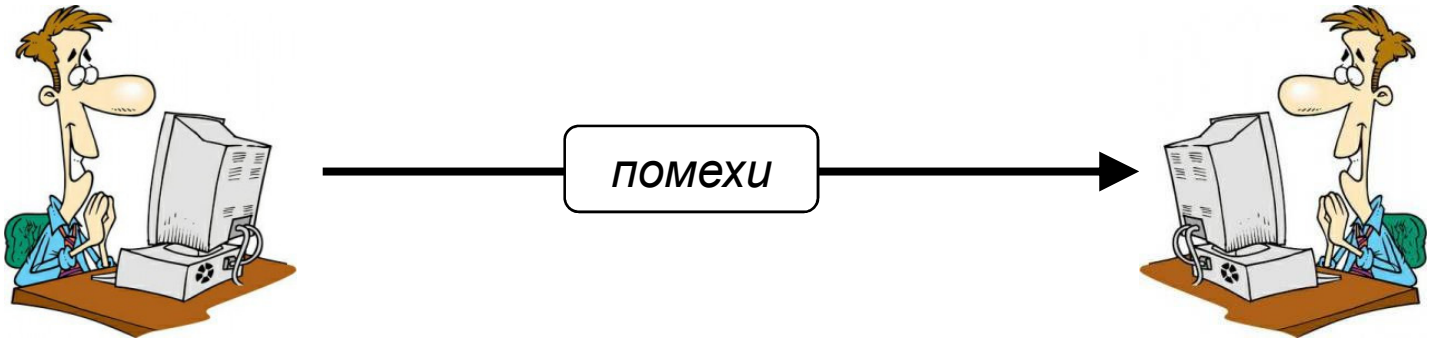
Путь есть помехи в канале связи и известна, например, вероятность их происхождения.

Например, известно, что если послать три бита подряд, то ошибка произойдет не более чем в одном из них.

Как можно поступить?

Кодирование информации и защита от помех

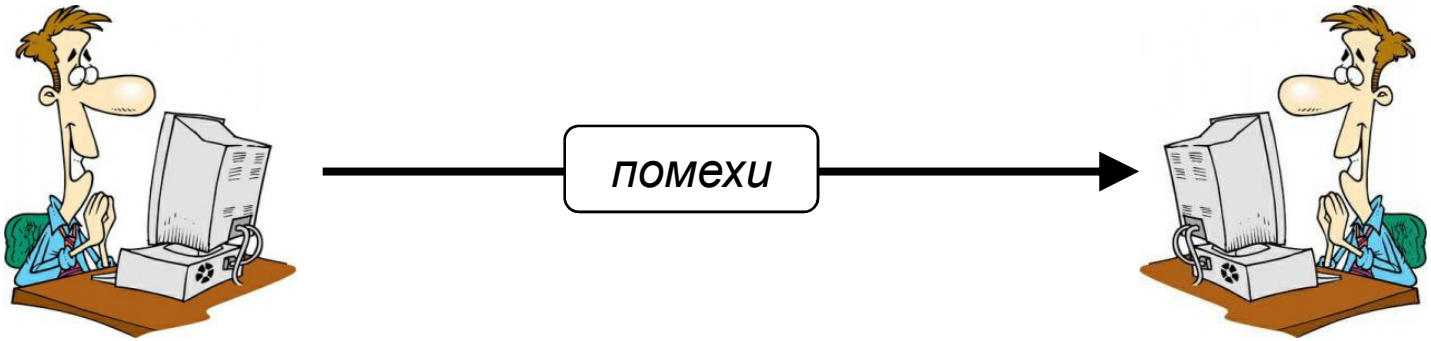
Задача: Надо отправить «да» или «нет» от одного пользователя к другому.



**Если послать три бита подряд,
то ошибка произойдет не
более чем в одном из них**

Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.

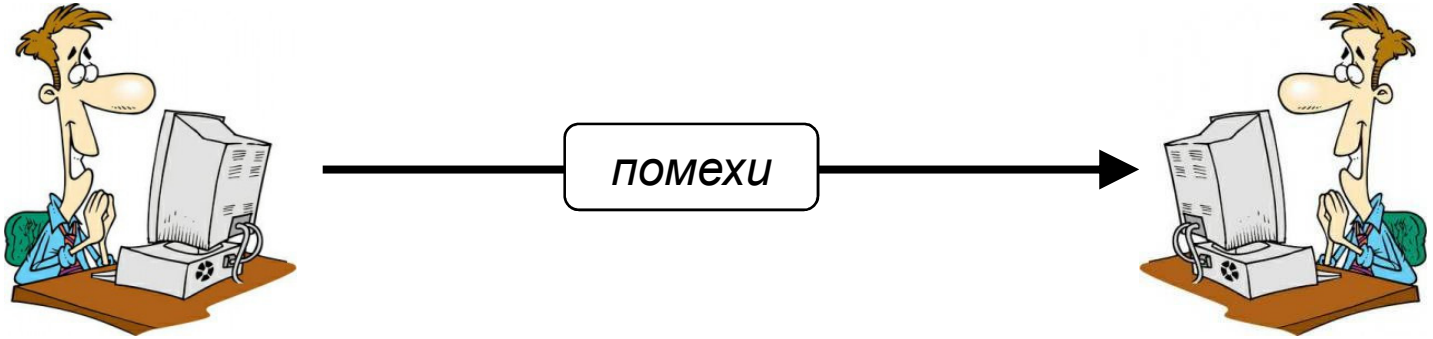


**Если послать три бита подряд,
то ошибка произойдет не
более чем в одном из них**

Решение:
Если «да», то посылаем 111
Если «нет», то посылаем 000

Кодирование информации и защита от помех

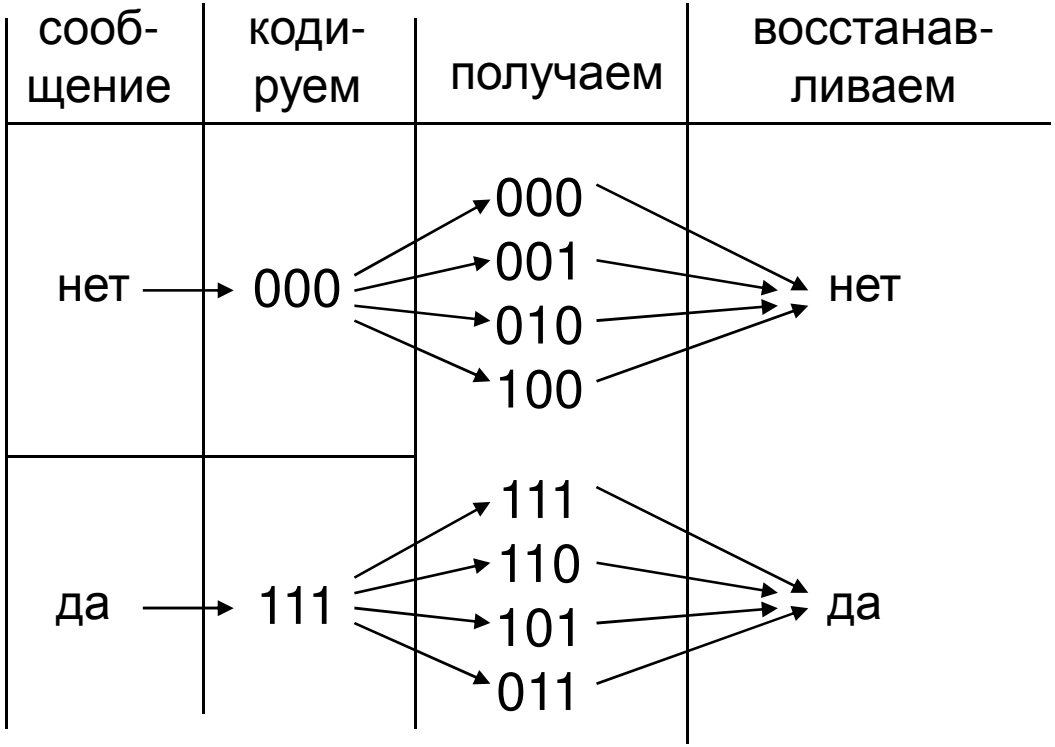
Задача: Надо отправить «да» или «нет» от одного пользователя к другому.



Если послать три бита подряд, то ошибка произойдет не более чем в одном из них

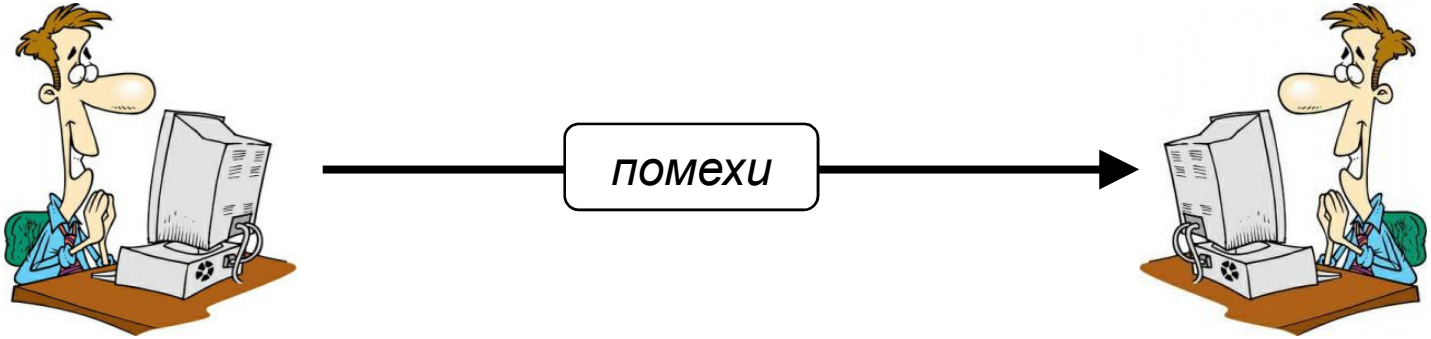
Решение:
Если «да», то посылаем 111
Если «нет», то посылаем 000

Почему это сработает?



Кодирование информации и защита от помех

Задача: Надо отправить «да» или «нет» от одного пользователя к другому.

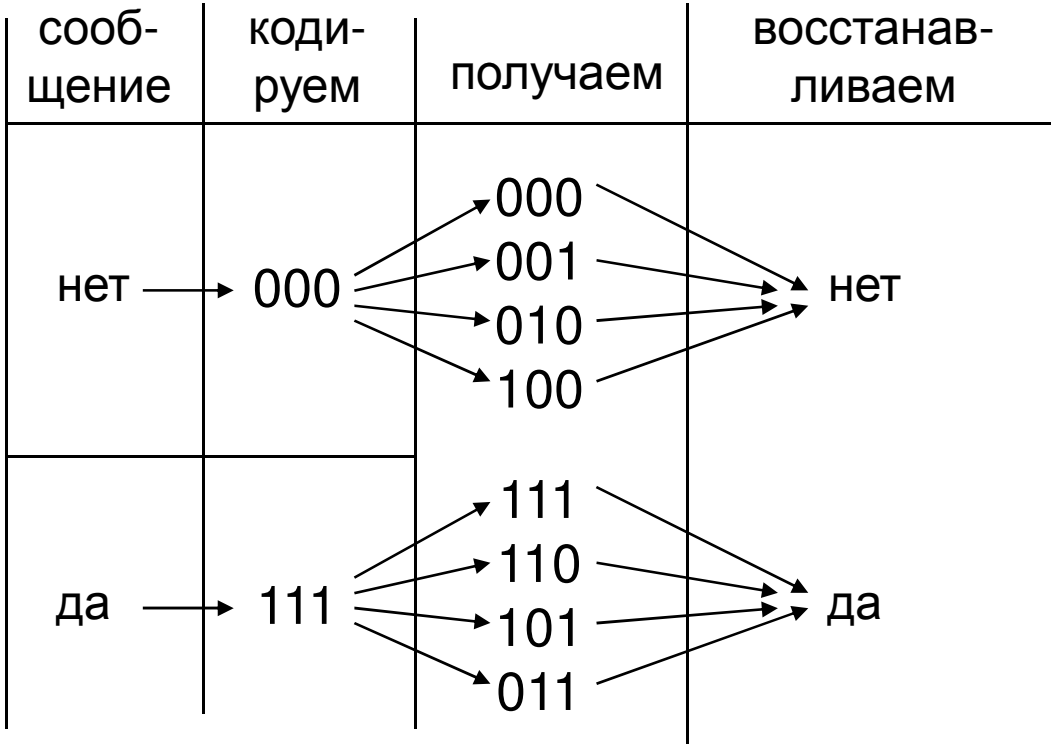


Если послать три бита подряд, то ошибка произойдет не более чем в одном из них

Решение:
Если «да», то посылаем 111
Если «нет», то посылаем 000

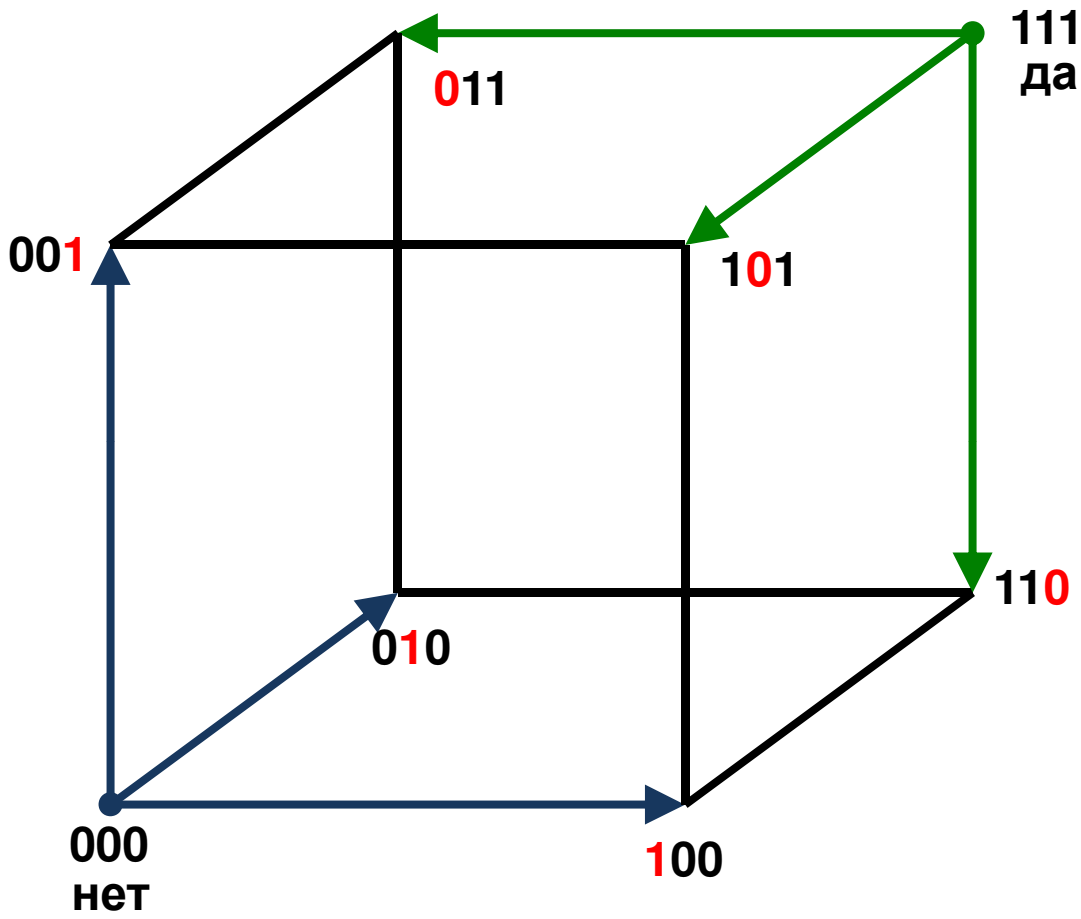
Почему это сработает?

А есть ли другие варианты?



Кодирование информации и защита от помех

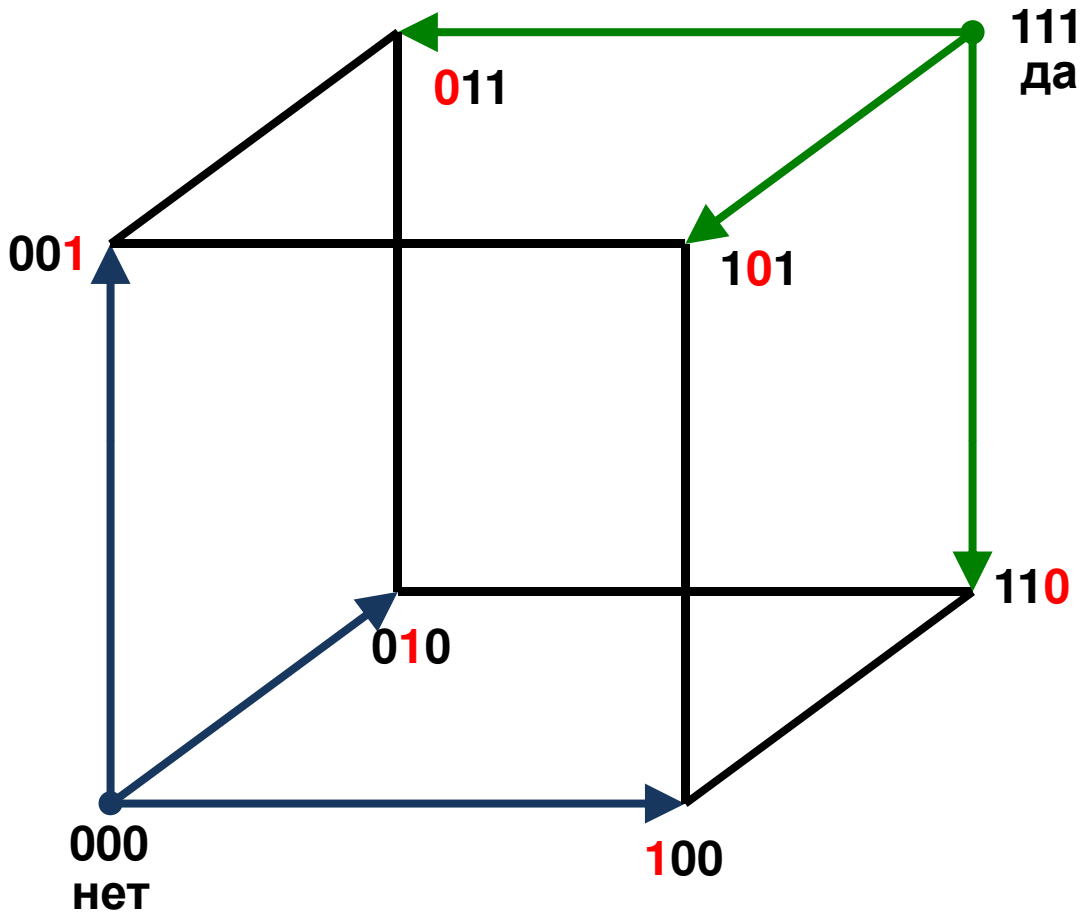
Геометрия кодирования на булевом кубе



Если «да», то посылаем 111
Если «нет», то посылаем 000

Кодирование информации и защита от помех

Геометрия кодирования на булевом кубе



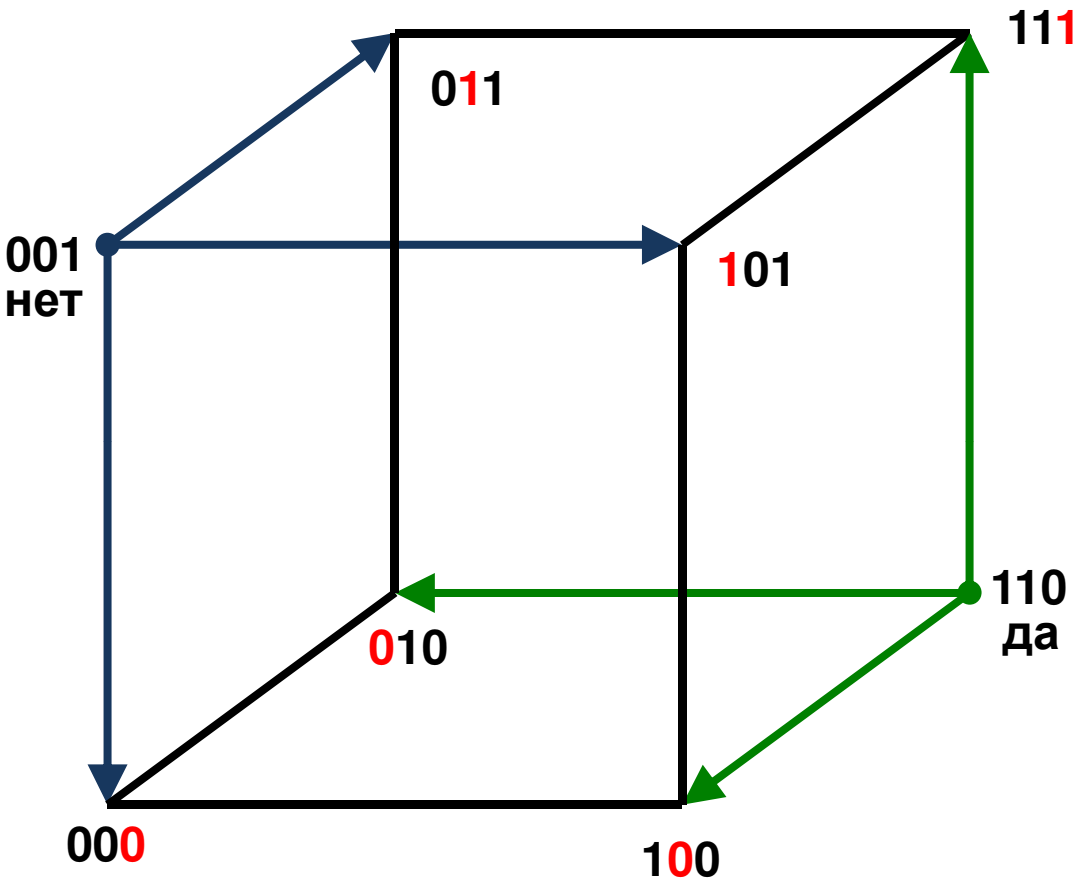
Если «да», то посылаем 111
Если «нет», то посылаем 000

А есть ли другие варианты?

Заметим, что картинка обладает симметрией.

Кодирование информации и защита от помех

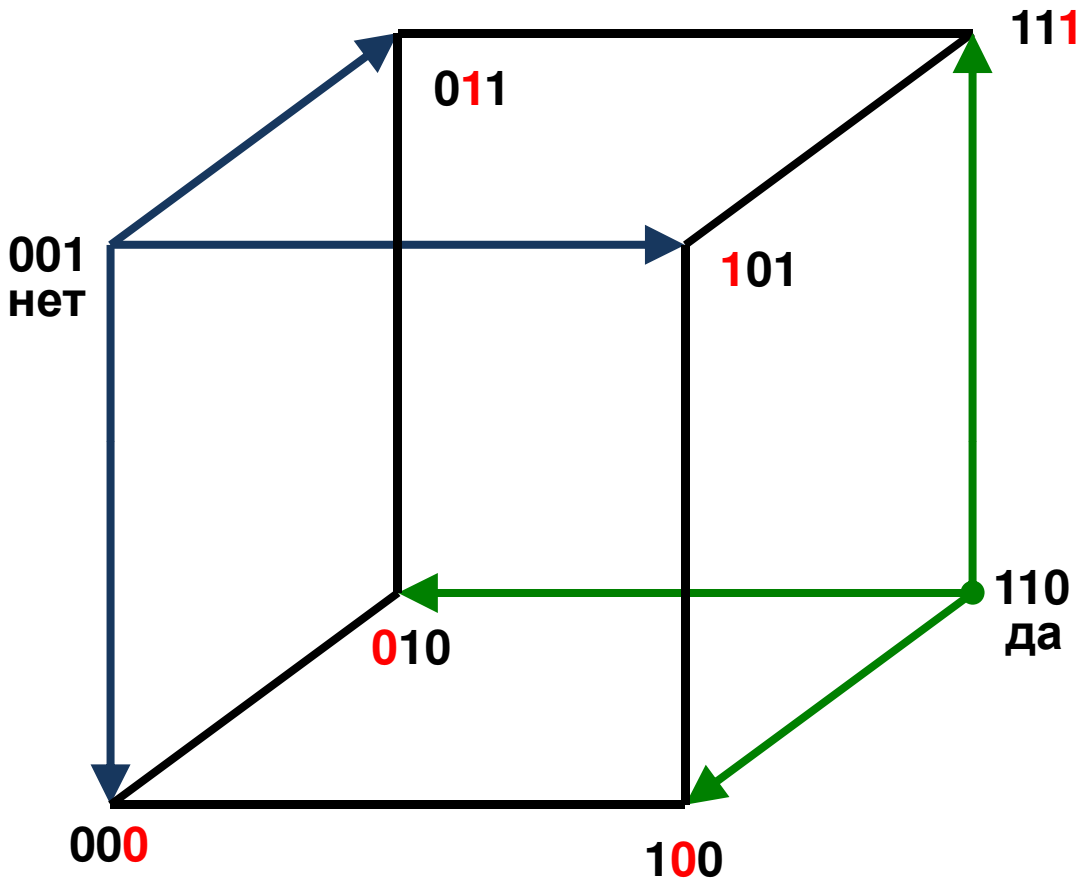
Геометрия кодирования на булевом кубе



Если «да», то посылаем 110
Если «нет», то посылаем 001

Кодирование информации и защита от помех

Геометрия кодирования на булевом кубе



Если «да», то посылаем 110
Если «нет», то посылаем 001

Геометрия 3-х мерного куба подсказала нам решение.

Но эта задача достаточно проста.

А многомерные кубы подскажут нам как решать куда как более сложные задачи…

Такие кубы называются Булевскими кубами.

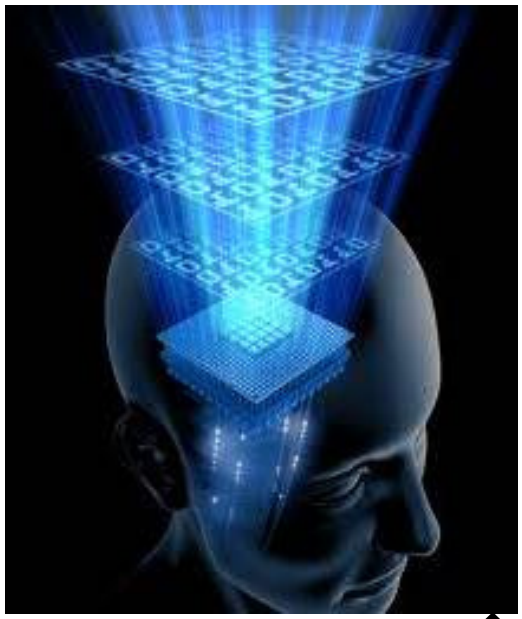
Часть 2
Алгоритмически неразрешимые проблемы

МГУ – ШКОЛЕ

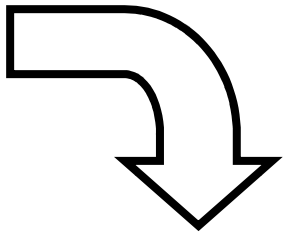
Часть 2
Алгоритмически неразрешимые проблемы

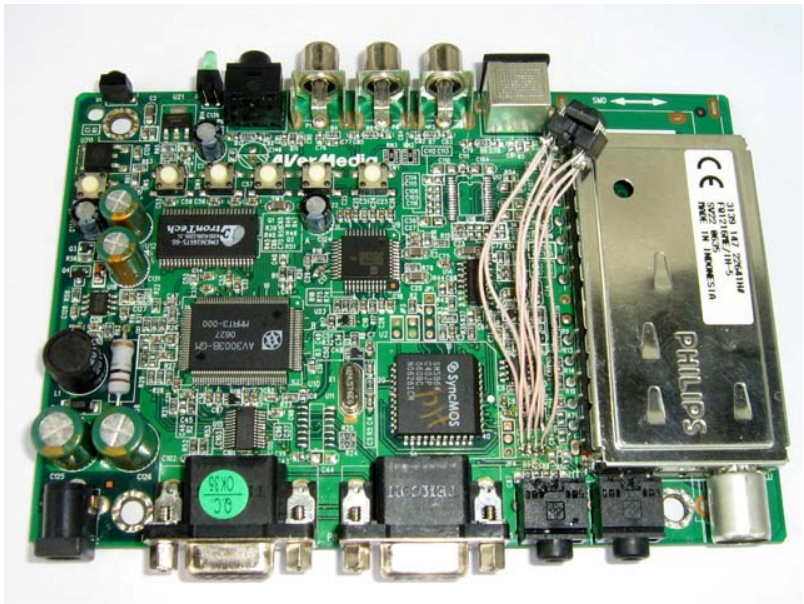
Разрушим миф об о всесильном искусственном интеллекте!

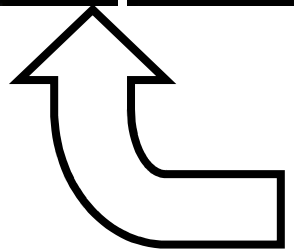
Как можно смотреть на компьютеры с точки зрения математики ?



Синтез схем, автоматов и алгоритмов

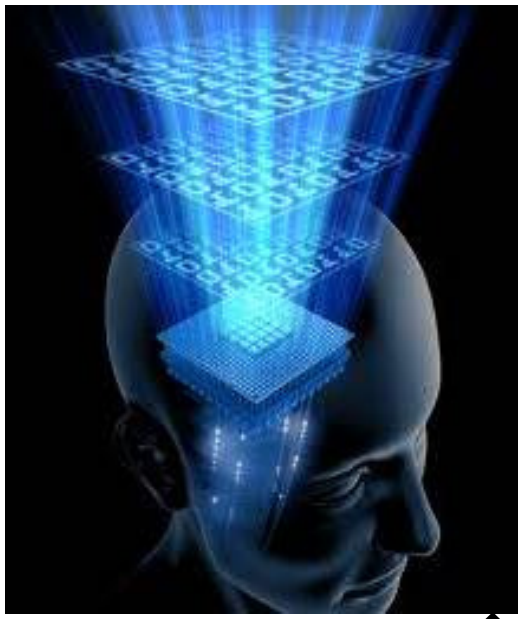






Математическая модель компьютера

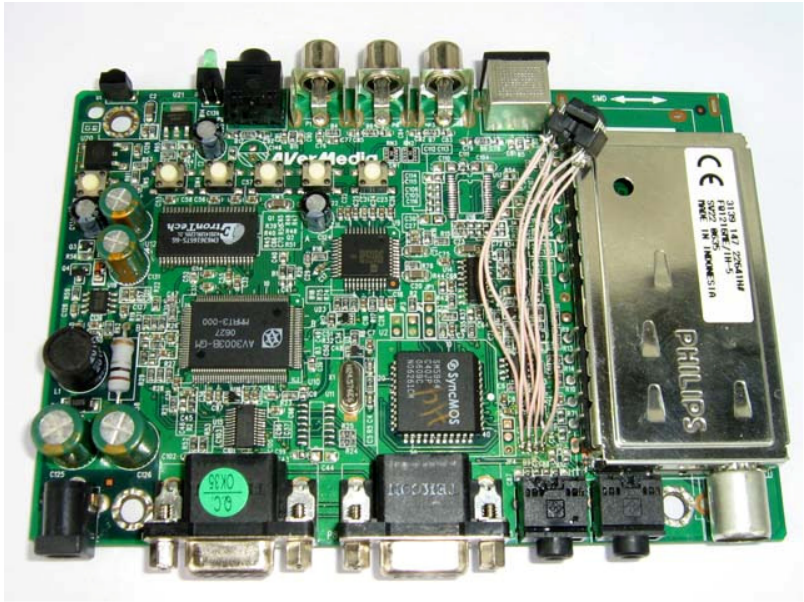
Как можно смотреть на компьютеры с точки зрения математики ?



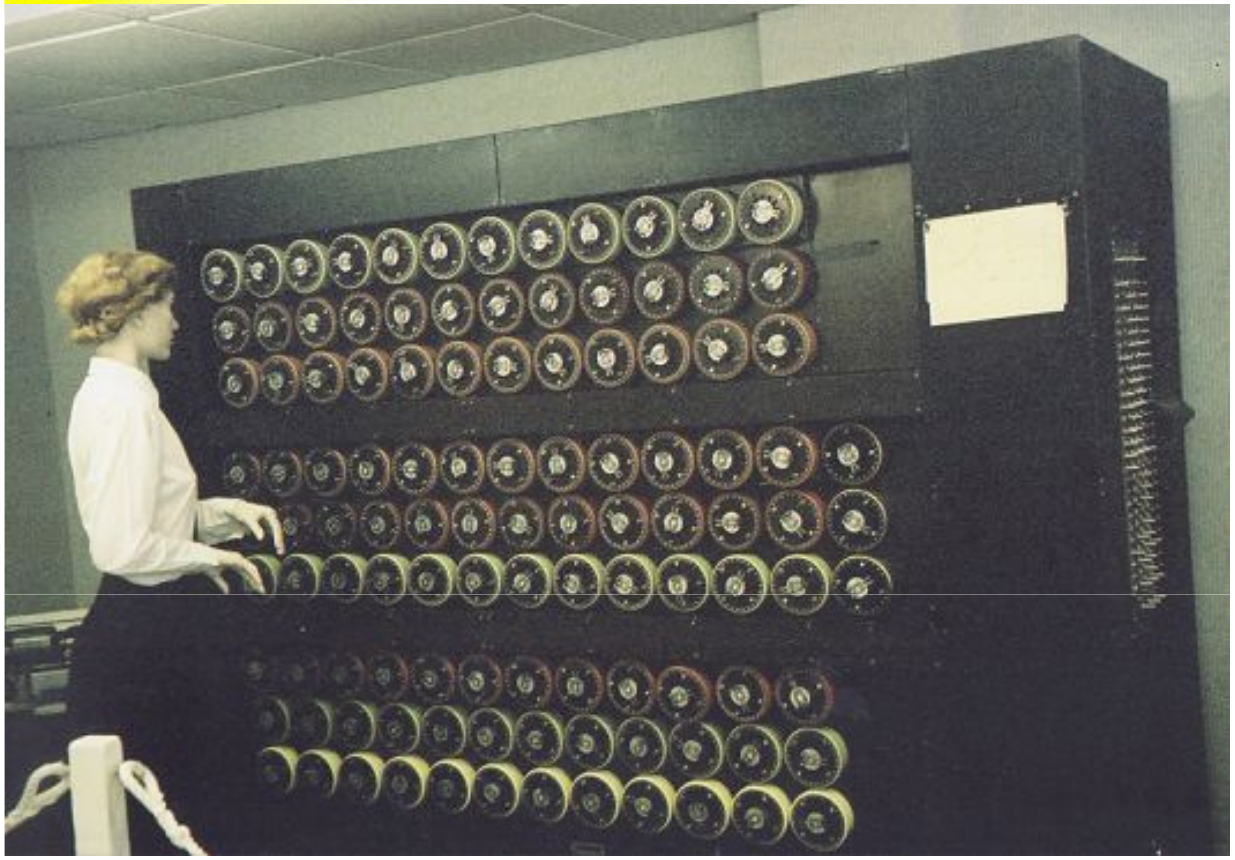
Синтез схем, автоматов и алгоритмов

Математическая модель компьютера

Важно, чтобы математическая модель была адекватной реальности!



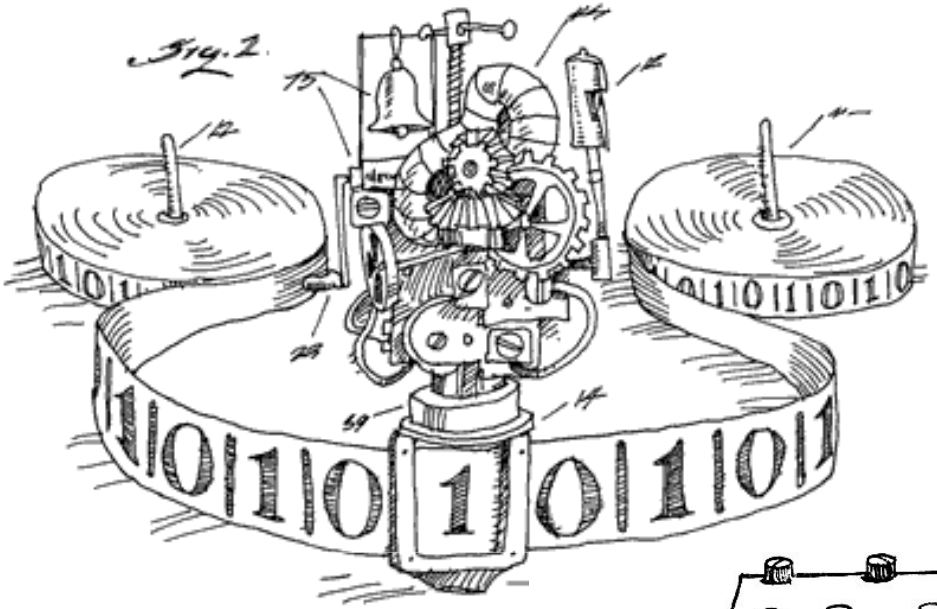
Машина Тьюринга

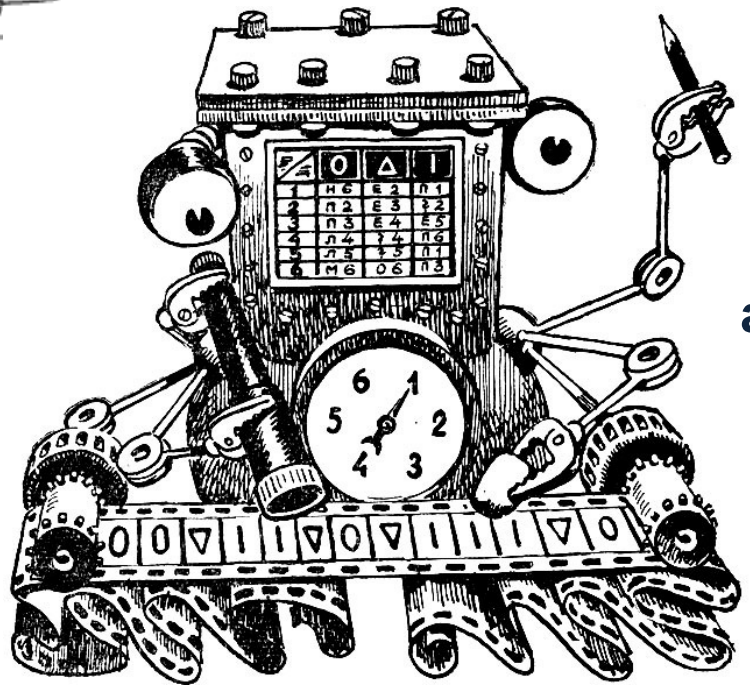


1936 год

Алан Тьюринг
(1912 - 1954)
английский математик,
логик, криптограф

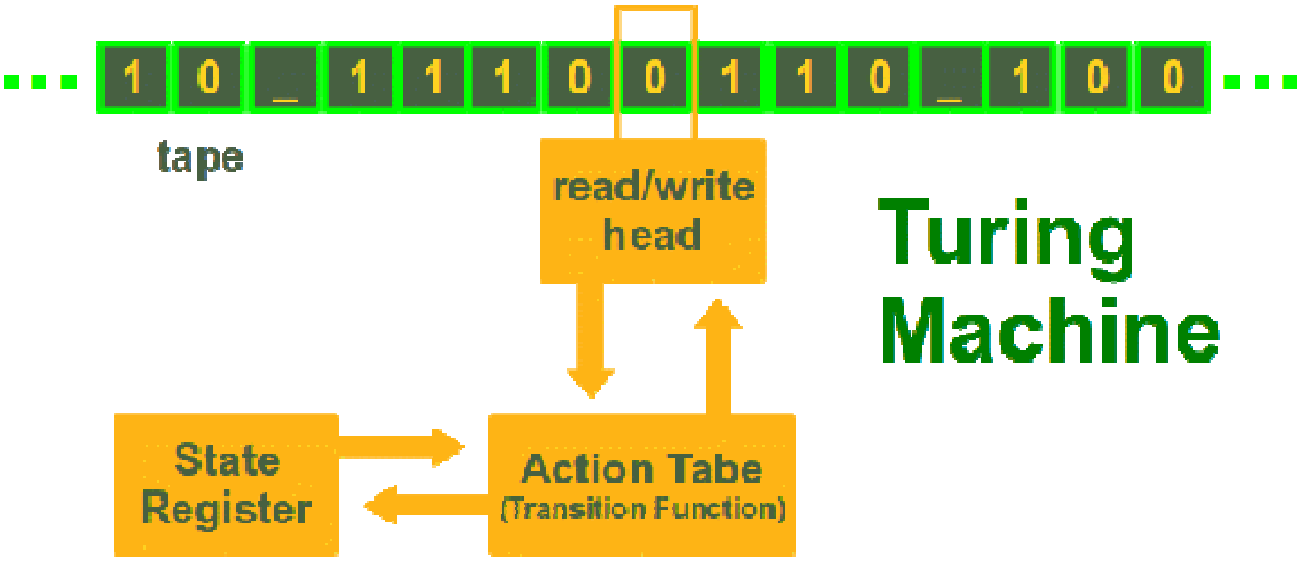
Машина Тьюринга





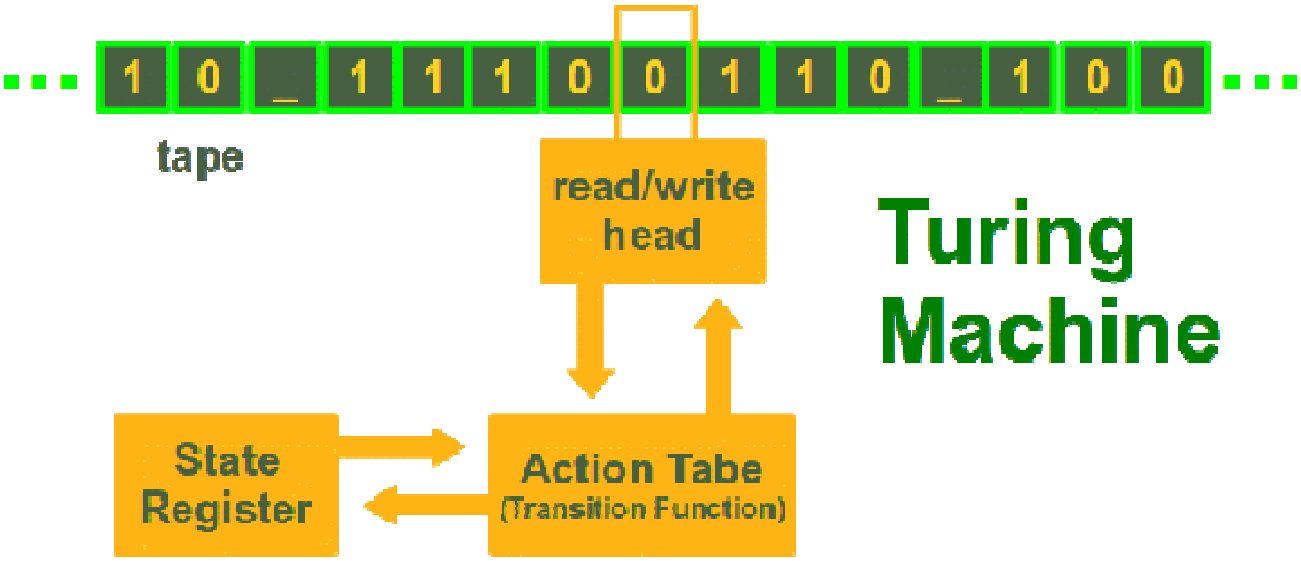
Алан Тьюринг
(1912 - 1954)
английский математик,
логик, криптограф

Машина Тьюринга



Алан Тьюринг
(1912 - 1954)
английский математик, логик, криптограф

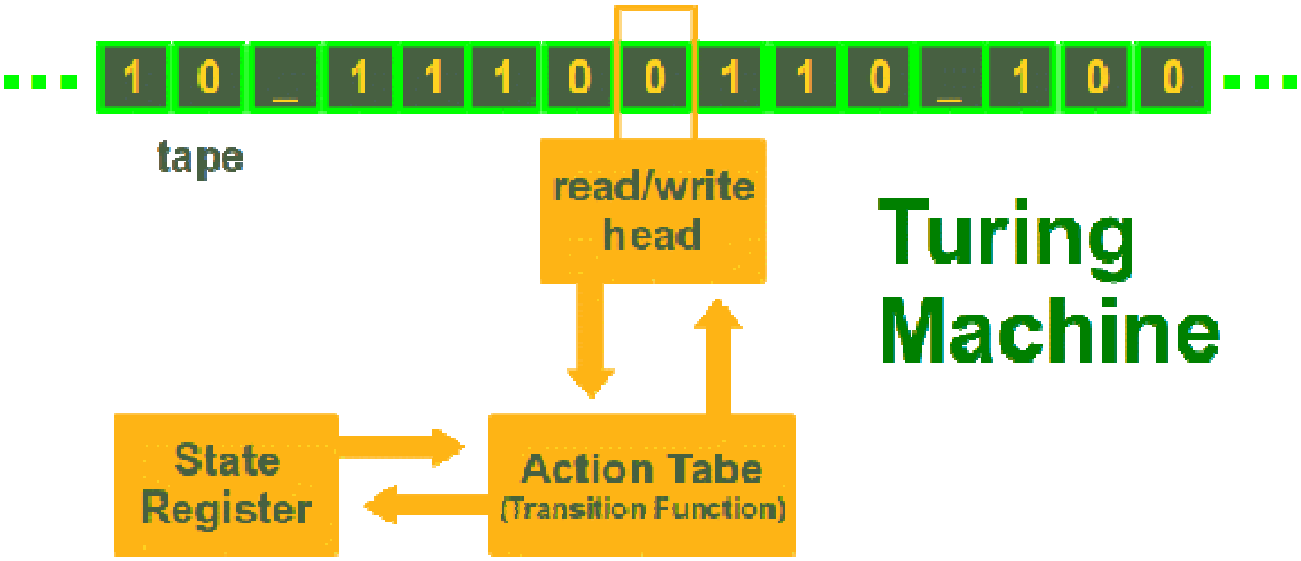
Машина Тьюринга



Посмотрим видео…

Алан Тьюринг
(1912 - 1954)
английский математик,
логик, криптограф

Машина Тьюринга



Алан Тьюринг
(1912 - 1954)
английский математик,
логик, криптограф

Команда:
Текущее состояние
Читаю с ленты
→
Записываю на ленту
Направление сдвига
Новое состояние

S_0 - начальное
S_0 0 →0 R S_0
S_0 1 →1 N S_1
S_1 0 →1 L S_2
S_1 1 →0 L S_1
S_2 0 →0 N S_3
S_2 1 →0 R S_1
S_3 – финальное

- это программа на машине Тюрина

В чем суть работы программы на компьютере?

Программа – форматизированный список инструкций работы с устройствами вводы-вывода данных и памятью.

Пользователь

Текст программы

Память компьютера

Развитие языков программирования

Программировать машину Тьюринга сложно.
Стали появляться разные языки программирования:
- процедурные: Basic, C, C++, Pascal, Fortran, Java, ...
- функциональные: Haskell, Lisp, ML, ...
...
- эзотерические (шуточные): Befunge, Piet ...

Вычисление факториала

По заданному числу n вычисляется число n! = 1·2·3·…(n-1) ·n

n	1	2	3	4	5
n!	1	2	6	24	120

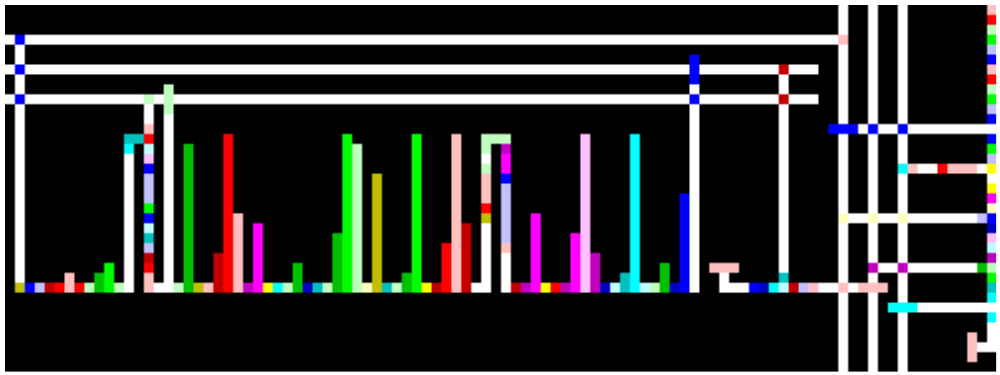
C – процедурный язык

```c
#include <stdio.h>
main()
{
 unsigned int n, i, x = 1;

printf("n = ");
 scanf("%i", &n);

for (i = 1; i <= n; i++) x *= i;
  printf("Result: %i", x);

 return 0;
}
```

Haskell – функциональный язык

```haskell
fac :: Integer -> Integer
fac 0 = 1
fac n | n > 0 = n * fac (n - 1)
```

Piet - эзотерический язык



Язык блок-схем

- Начало программы

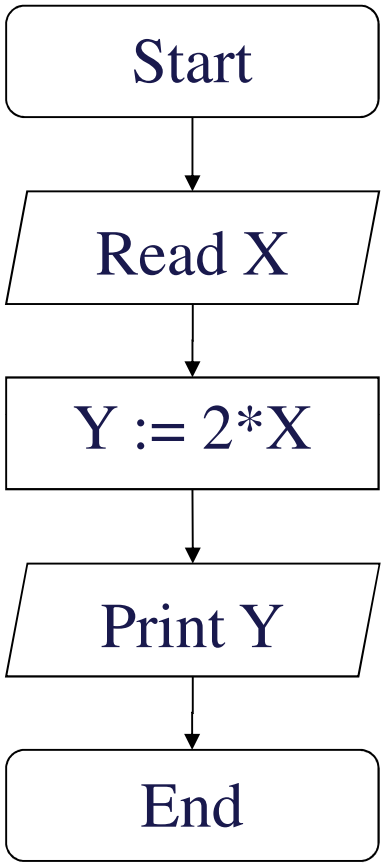
- Конец программы

- Чтение информации в переменную X

- Печать информации из переменной X

Примеры блок-схем

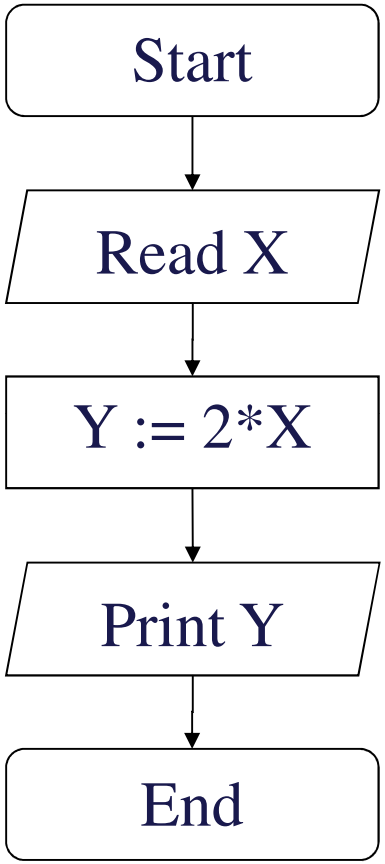
P_1 - вычисление 2*x



$P_1(1) = 2$
$P_1(2) = 4$

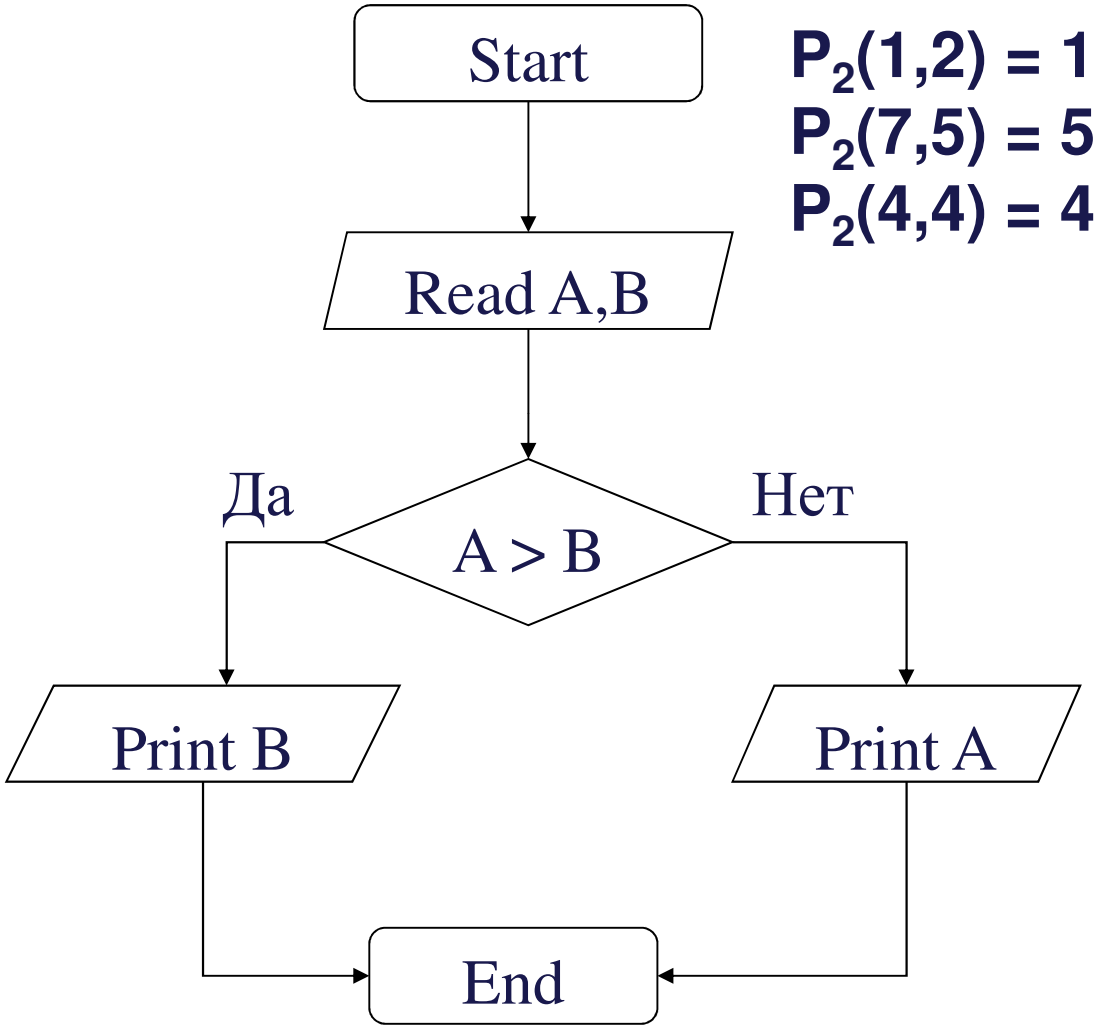
Примеры блок-схем

P_1 - вычисление 2*x



$P_1(1) = 2$
$P_1(2) = 4$

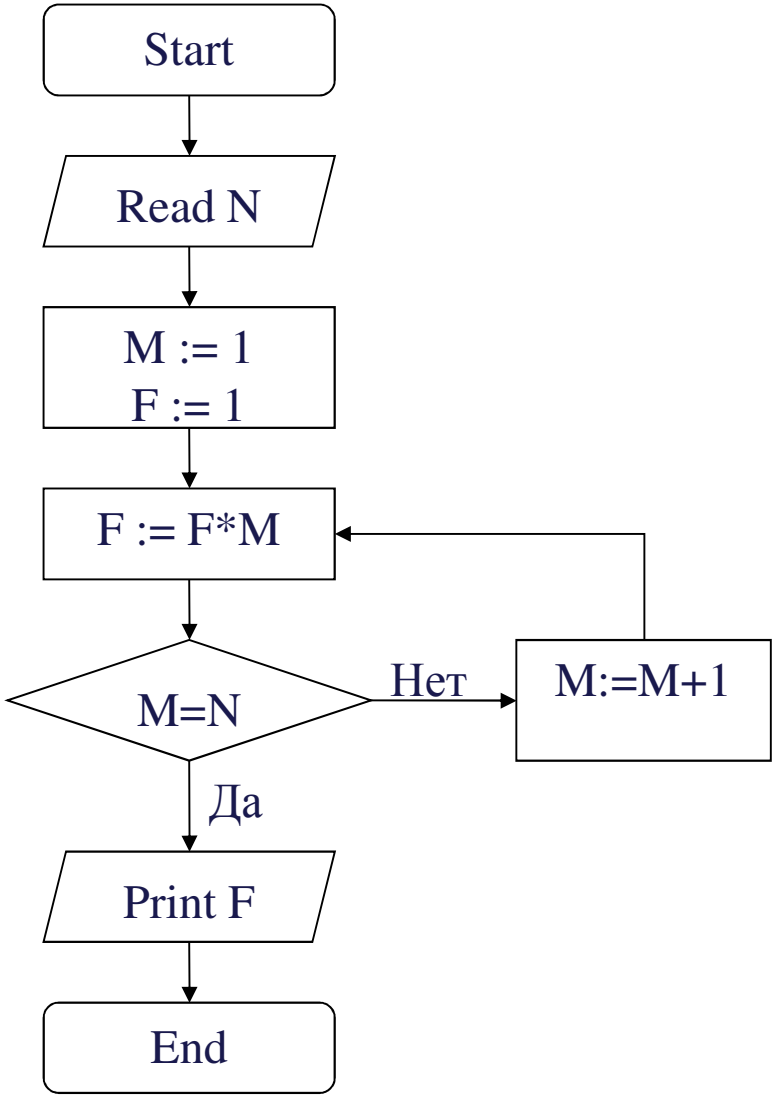
P_2 – вычисление минимума (a,b)



$P_2(1,2) = 1$
$P_2(7,5) = 5$
$P_2(4,4) = 4$

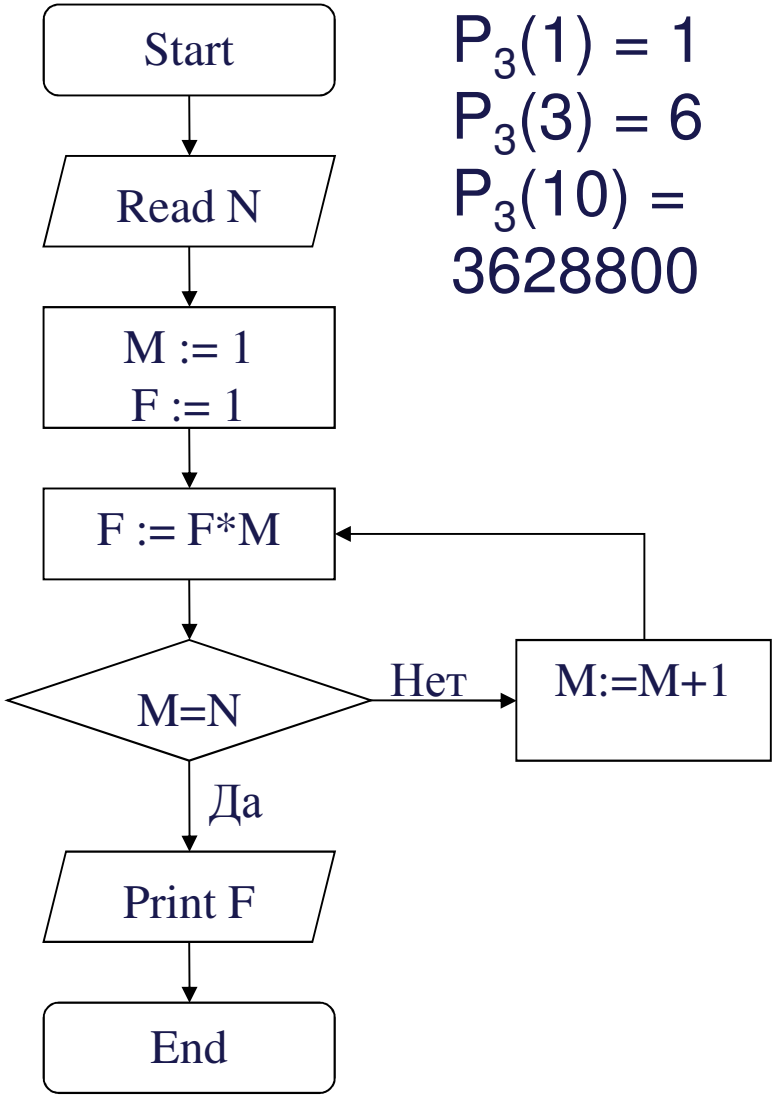
Примеры блок-схем

P_3 - вычисление n!



Примеры блок-схем

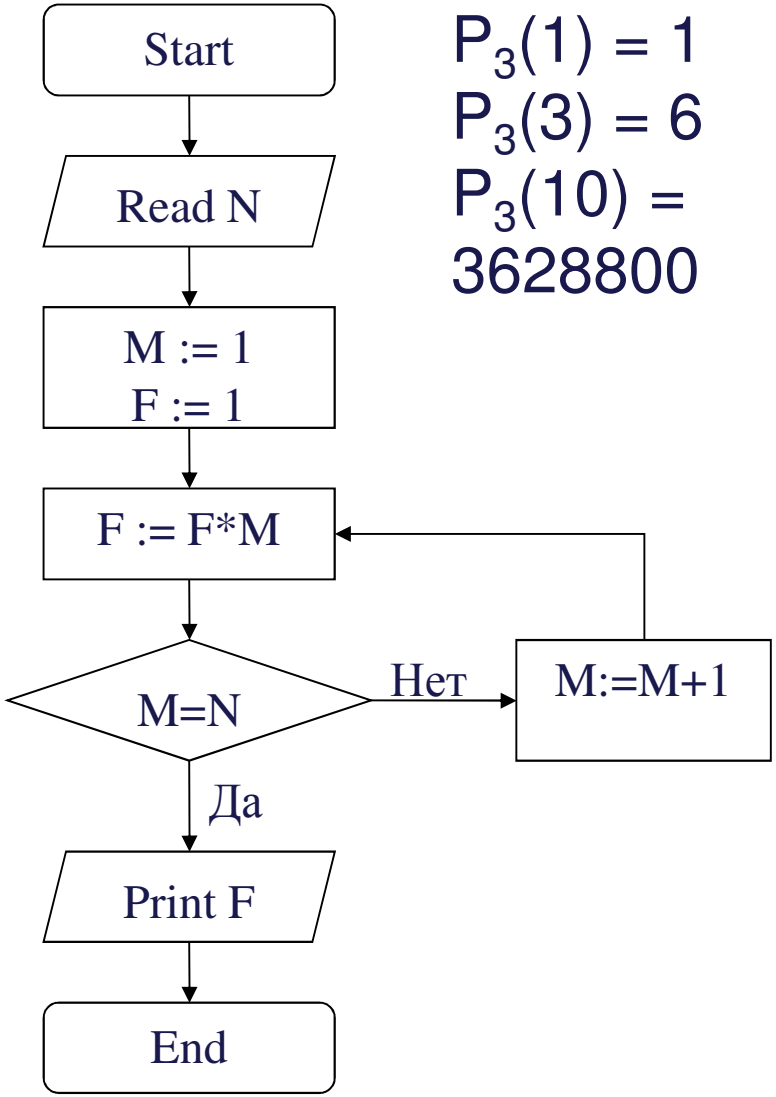
P_3 - вычисление n!



$P_3(1) = 1$
$P_3(3) = 6$
$P_3(10) = 3628800$

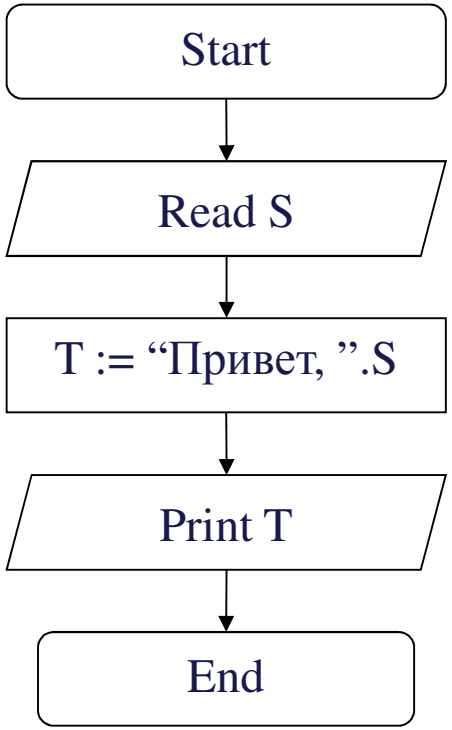
Примеры блок-схем

P_3 - вычисление n!



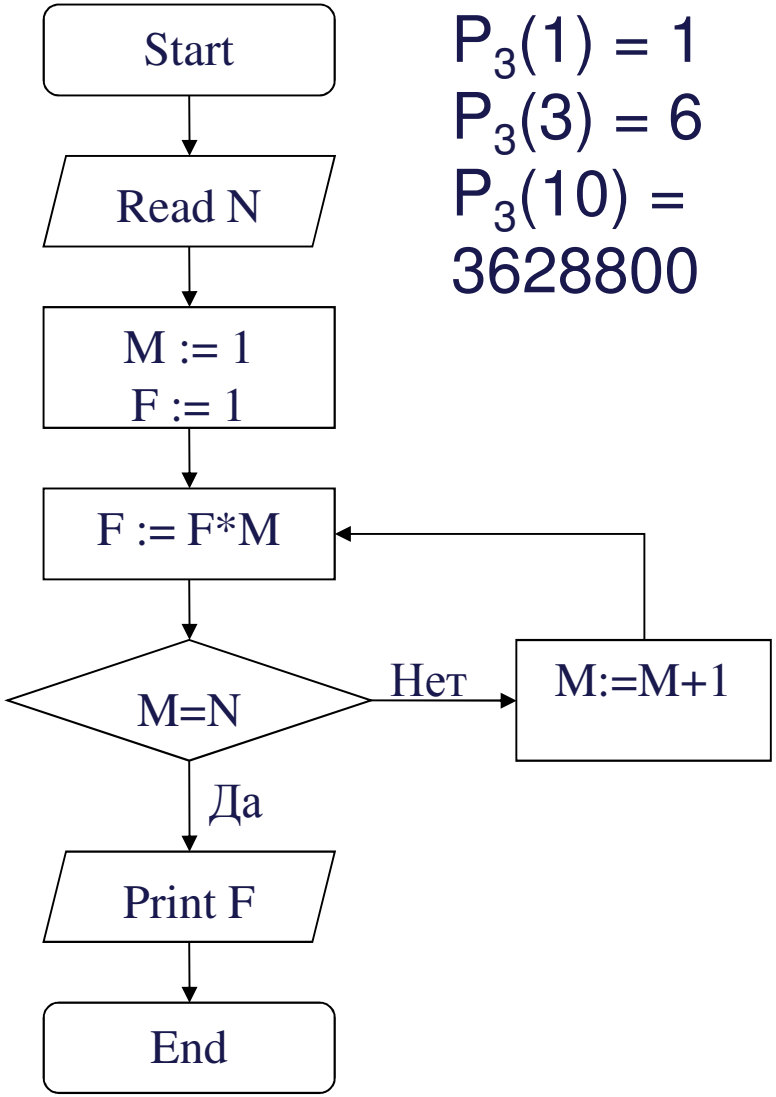
$P_3(1) = 1$
$P_3(3) = 6$
$P_3(10) = 3628800$

P_4 - приветствие пользователя



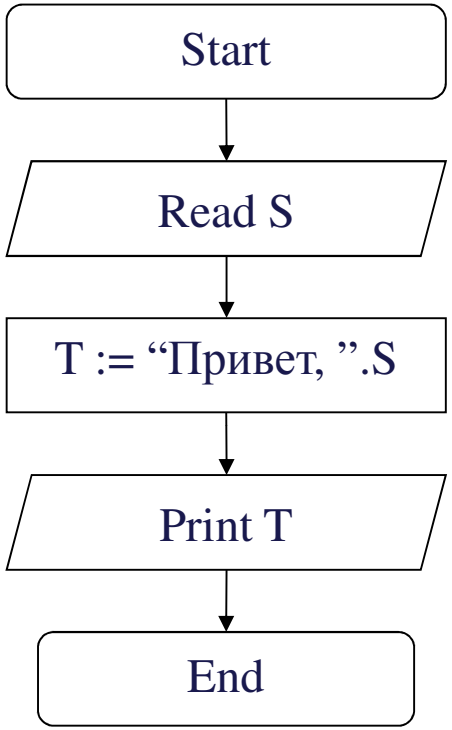
Примеры блок-схем

P_3 - вычисление n!



$P_3(1) = 1$
$P_3(3) = 6$
$P_3(10) = 3628800$

P_4 - приветствие пользователя



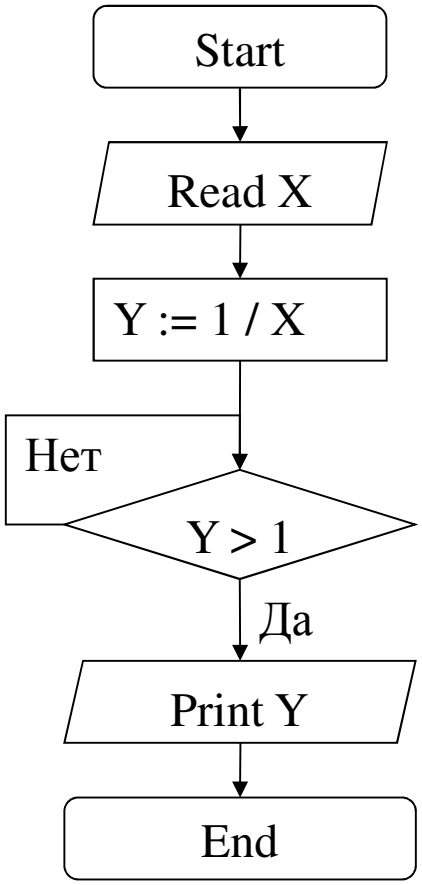
P_4(“Вася”) = “Привет, Вася”
P_4(“Петя”) = “Привет, Петя”
P_4(“Привет”) = “Привет, Привет”

МГУ – ШКОЛЕ

«Плохие» программы

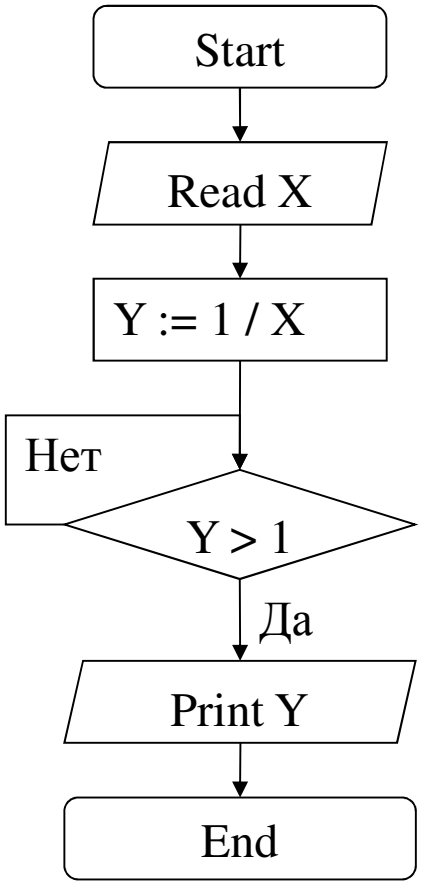
«Плохие» программы

P_5 - «плохая» программа



«Плохие» программы

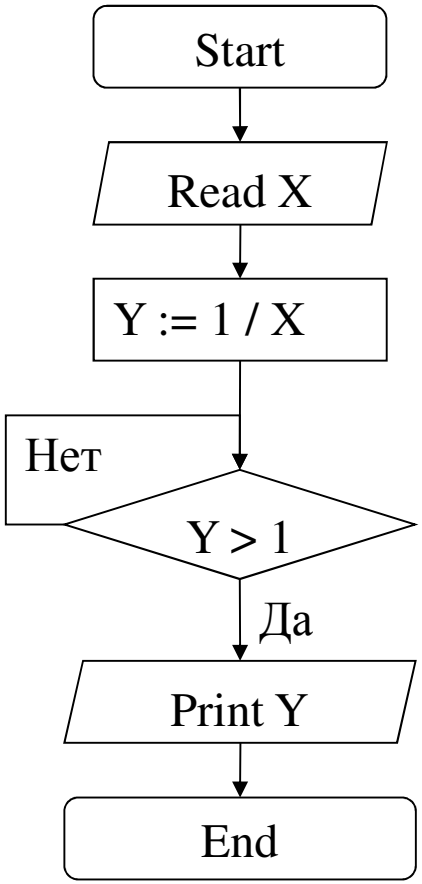
P_5 - «плохая» программа



$P_5(0.5) = 2$

«Плохие» программы

P_5 - «плохая» программа

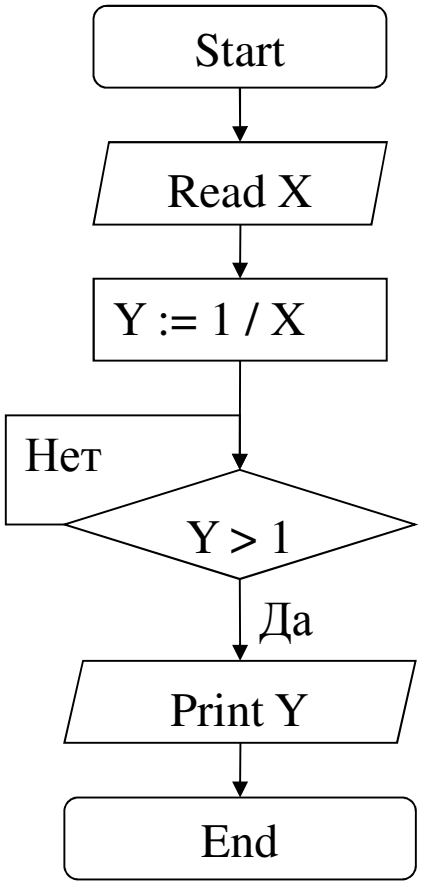


$P_5(0.5) = 2$

$P_5(0)$ – ошибка

«Плохие» программы

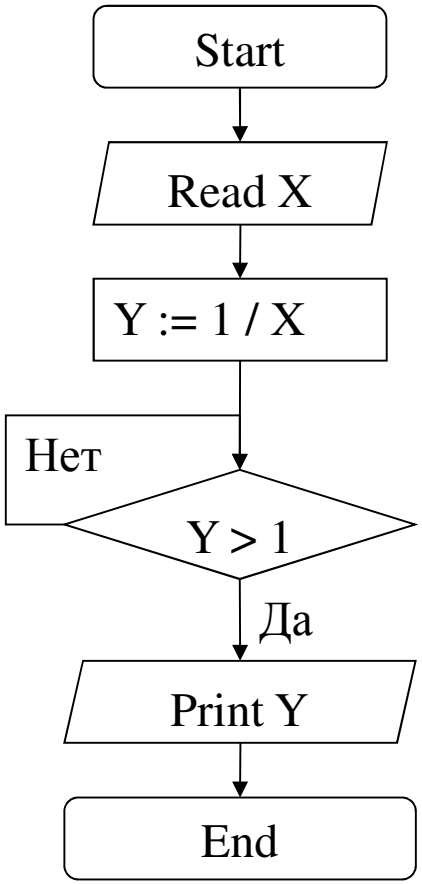
P_5 - «плохая» программа



$P_5(0.5) = 2$
$P_5(0)$ – ошибка
$P_5(2)$ – зависает

«Плохие» программы

P_5 - «плохая» программа



$P_5(0.5) = 2$
$P_5(0)$ – ошибка
$P_5(2)$ – зависает

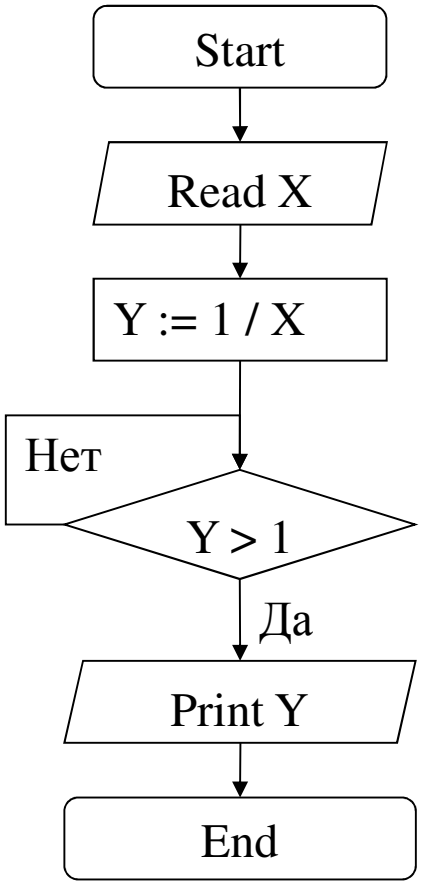
Если X=0, то ошибка

Если 0<X<1, то завершает

Иначе – зависает

«Плохие» программы

P_5 - «плохая» программа



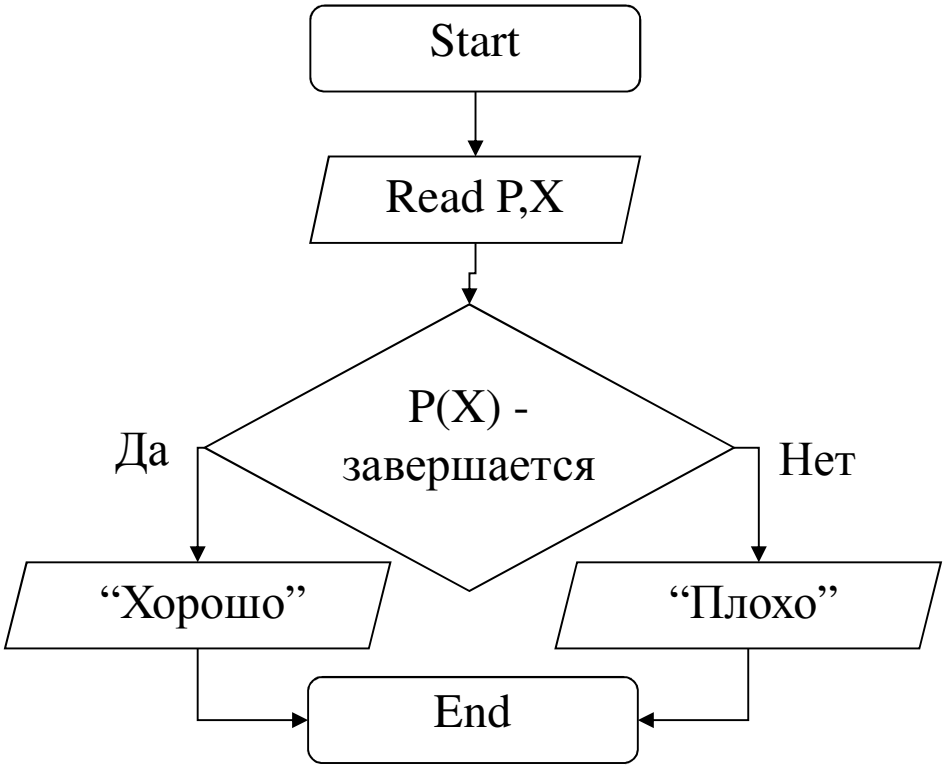
$P_5(0.5) = 2$
$P_5(0)$ – ошибка
$P_5(2)$ – зависает

Если X=0, то ошибка

Если 0<X<1, то завершает

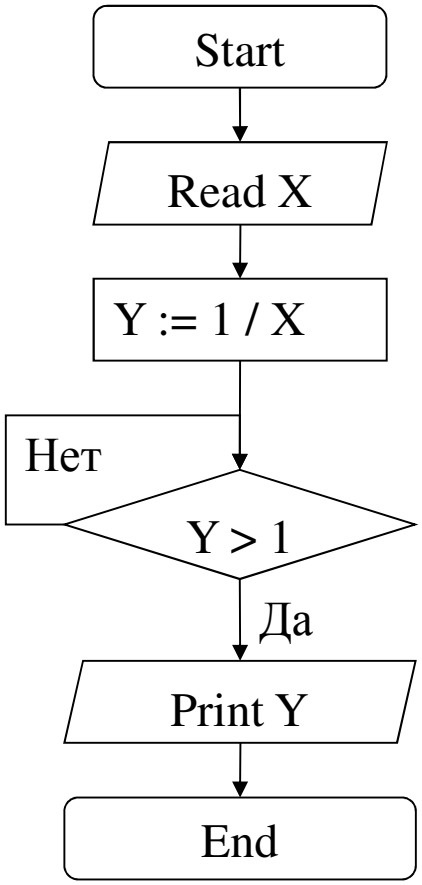
Иначе – зависает

P_6 - проверка завершения работы



«Плохие» программы

P_5 - «плохая» программа



$P_5(0.5) = 2$
$P_5(0)$ – ошибка
$P_5(2)$ – зависает

Если X=0, то ошибка

Если 0<X<1, то завершает

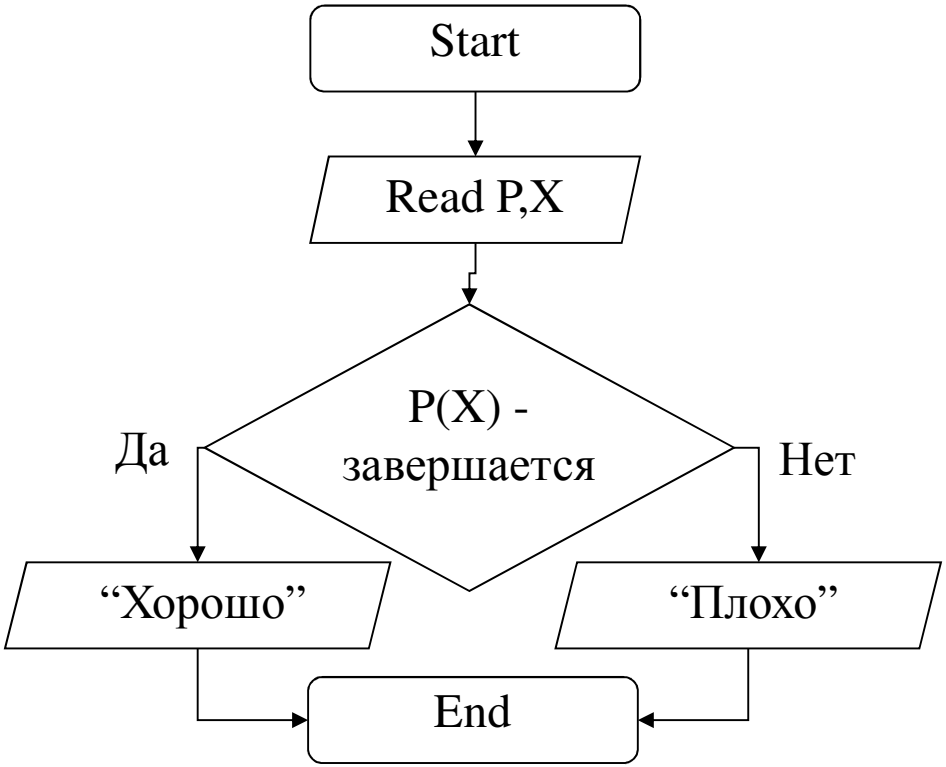
Иначе – зависает

P_6 - проверка завершения работы

$P_6(P_5,0.5)$ = “Хорошо”

$P_6(P_5,0)$ = “Плохо”

$P_6(P_5,2)$ = “Плохо”



Алгоритмически неразрешимые задачи

Теорема:

- Проблема завершения работы программы алгоритмически неразрешима

Алгоритмически неразрешимые задачи

Теорема:

- Проблема завершения работы программы алгоритмически неразрешима

или

- Проблема проверки программы на то, что она зависла алгоритмически неразрешима

Алгоритмически неразрешимые задачи

Теорема:

- Проблема завершения работы программы алгоритмически неразрешима

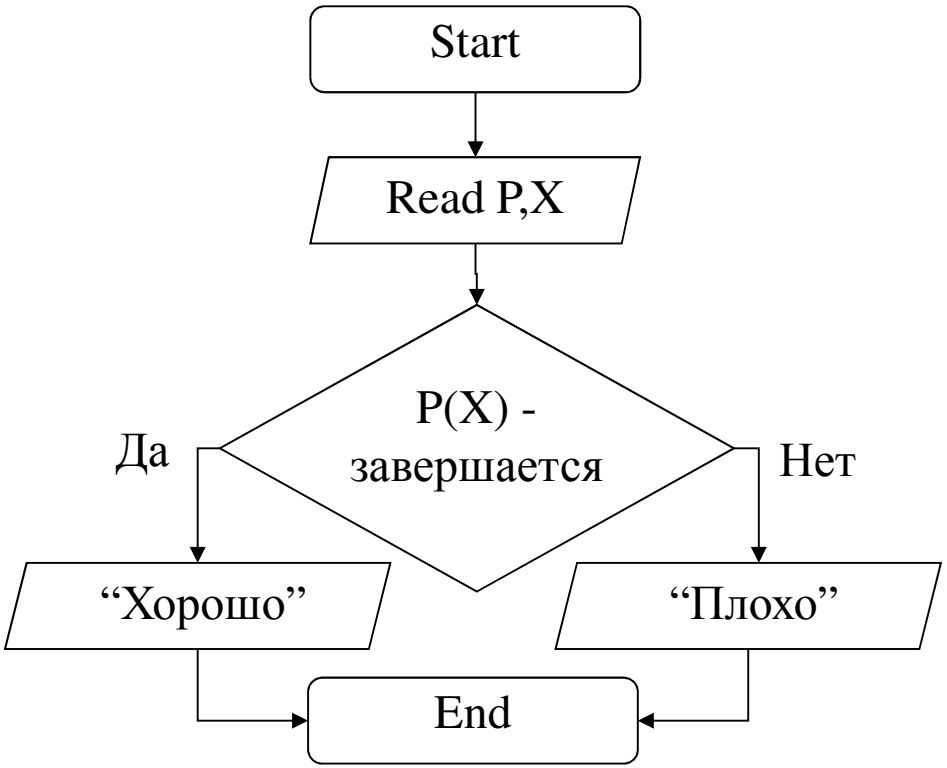
или

- Проблема проверки программы на то, что она зависла алгоритмически неразрешима

или

- Программу вида P_6 написать невозможно

P_6 - проверка завершения работы



Алгоритмически неразрешимые задачи

Теорема:

- Проблема завершения работы программы алгоритмически неразрешима

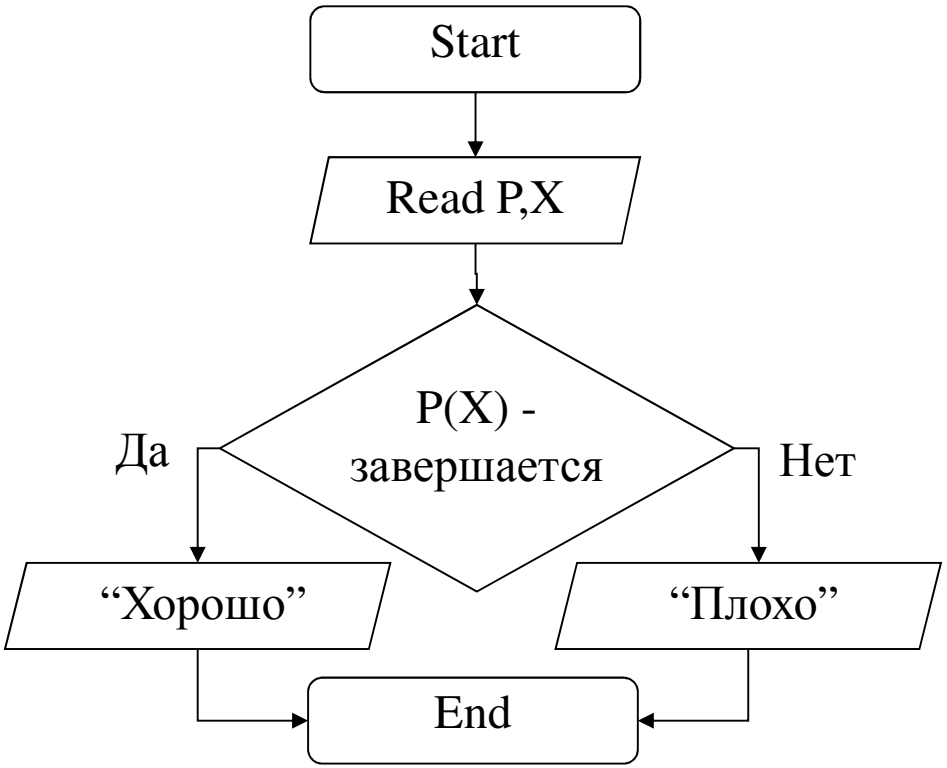
или

- Проблема проверки программы на то, что она зависла алгоритмически неразрешима

или

- Программу вида P_6 написать невозможно

P_6 - проверка завершения работы



- А как же диспетчер задач !?

Алгоритмически неразрешимые задачи

Теорема:

- Проблема завершения работы программы алгоритмически неразрешима

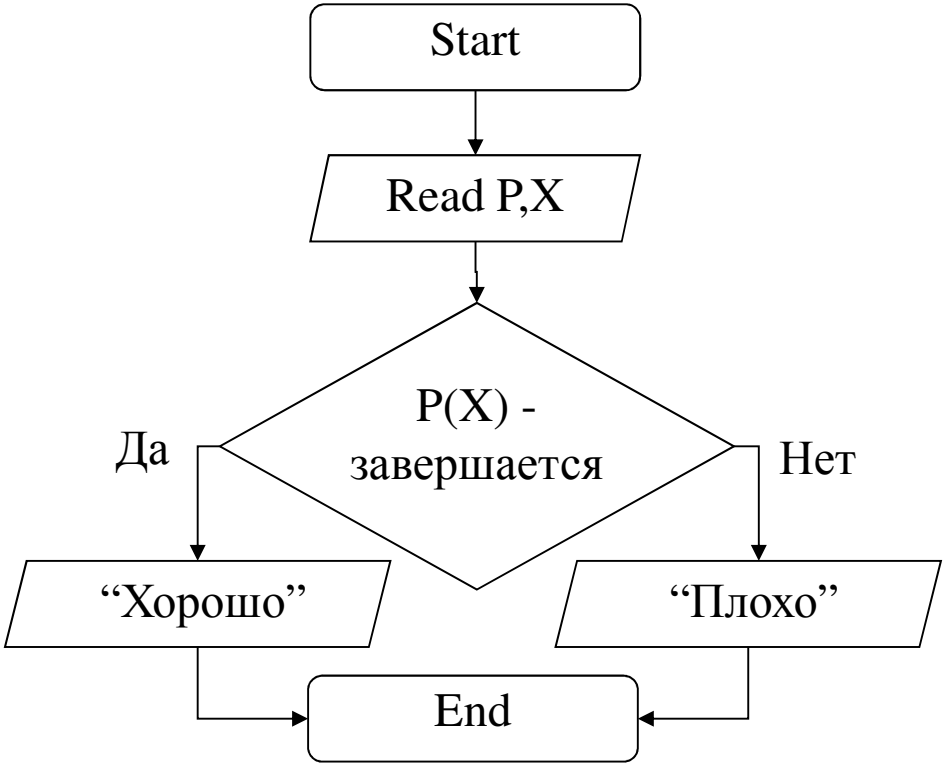
или

- Проблема проверки программы на то, что она зависла алгоритмически неразрешима

или

- Программу вида P_6 написать невозможно

P_6 - проверка завершения работы



- А как же диспетчер задач !?
- Он врет, или не это имеет в виду

Самоприменимые программы

Определение:

Программа называется **самоприменимой**,
если она завершает работу или не вызывает ошибки
при подаче ей на вход собственного кода.

Программа называется **не самоприменимой**,
если она зависает или вызывает ошибку
при подаче ей на вход собственного кода.

Самоприменимые программы

Определение:

Программа называется **самоприменимой**,
если она завершает работу или не вызывает ошибки
при подаче ей на вход собственного кода.

Программа называется **не самоприменимой**,
если она зависает или вызывает ошибку
при подаче ей на вход собственного кода.

Пример:
Блокнот (notepad.exe) – самоприменимая
Paint (mspaint.exe) – не самоприменимая

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Проверка на самоприменимость

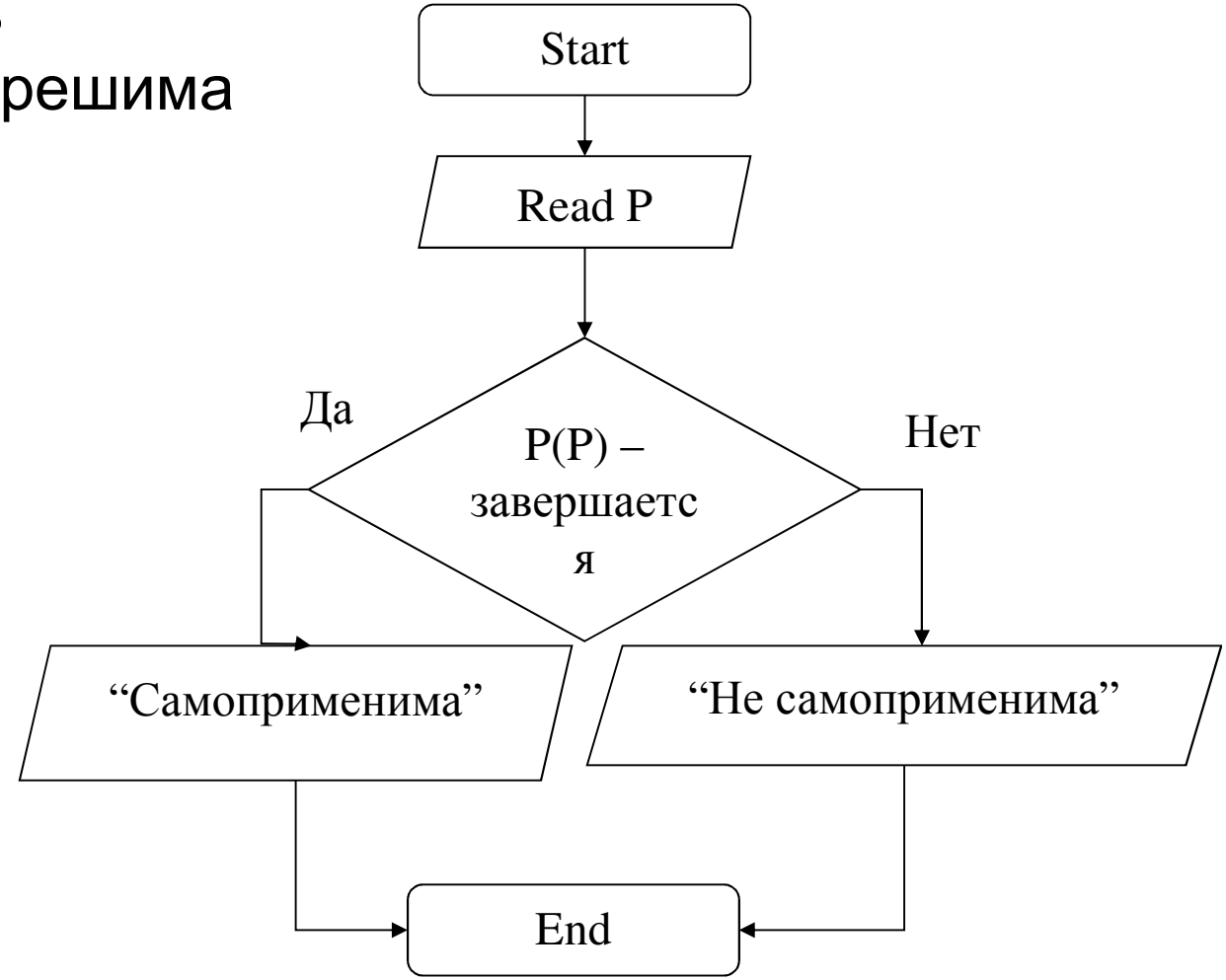
Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

или

- Программу вида P_7 написать невозможно

P_7 - проверка на самоприменимость



Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

То, что написано на табличке правда или ложь?

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

То, что написано на табличке правда или ложь?

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.

Парадокс брадобрея:
Единственному деревенскому брадобрею приказали в деревне: *«брить всякого, кто сам не бреется, и не брить того, кто сам бреется»*. Кто побреет брадобрея?

Проверка на самоприменимость

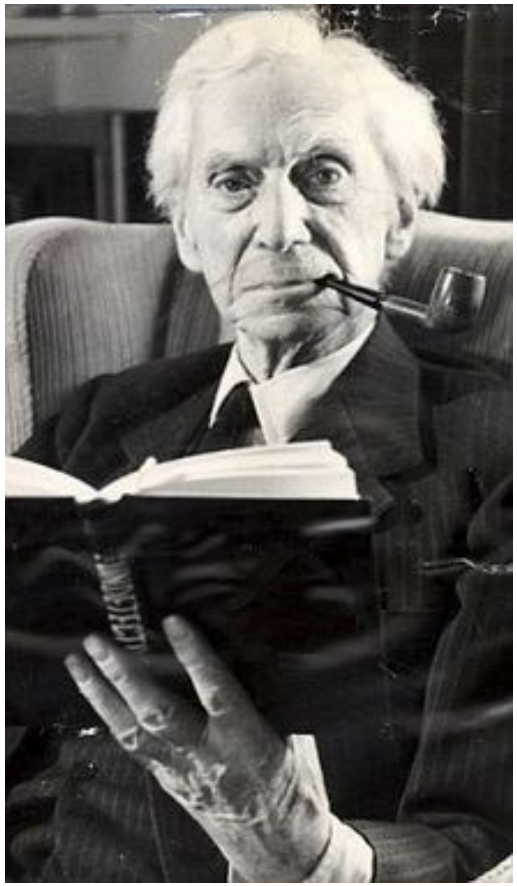
Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.



Бертран Артур Уильям Рассел
английский математик, философ
и общественный деятель

Парадокс Бертрана Рассела:
Пусть К — множество всех множеств, которые не содержат себя в качестве своего элемента.
Содержит ли К само себя в качестве элемента?

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

То, что написано на табличке правда или ложь?

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.

- Так какая же она ?!

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

То, что написано на табличке правда или ложь?

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.

- Так какая же она ?!

- Да не важно! Отстаньте от меня с глупыми вопросами!

... ответил человек ...

Проверка на самоприменимость

Лемма:

- Проблема проверки программы на самоприменимость алгоритмически неразрешима

Идея доказательства:

Это утверждение ложно

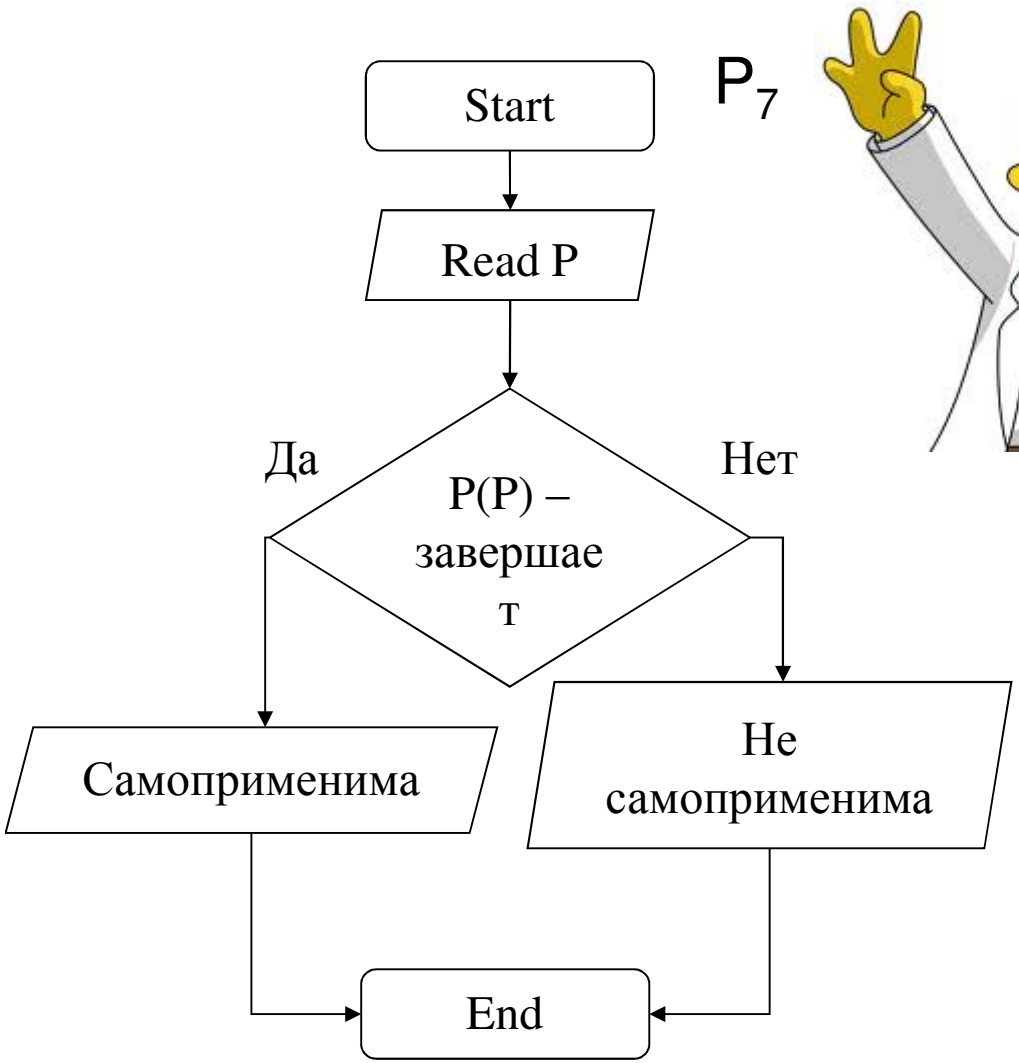
То, что написано на табличке правда или ложь?

Если надпись правда, то она ложь.
Если надпись ложь, то она правда.

- Так какая же она ?!
- Да не важно! Отстаньте от меня с глупыми вопросами!
... ответил человек ...
А компьютер ?

Доказательство леммы

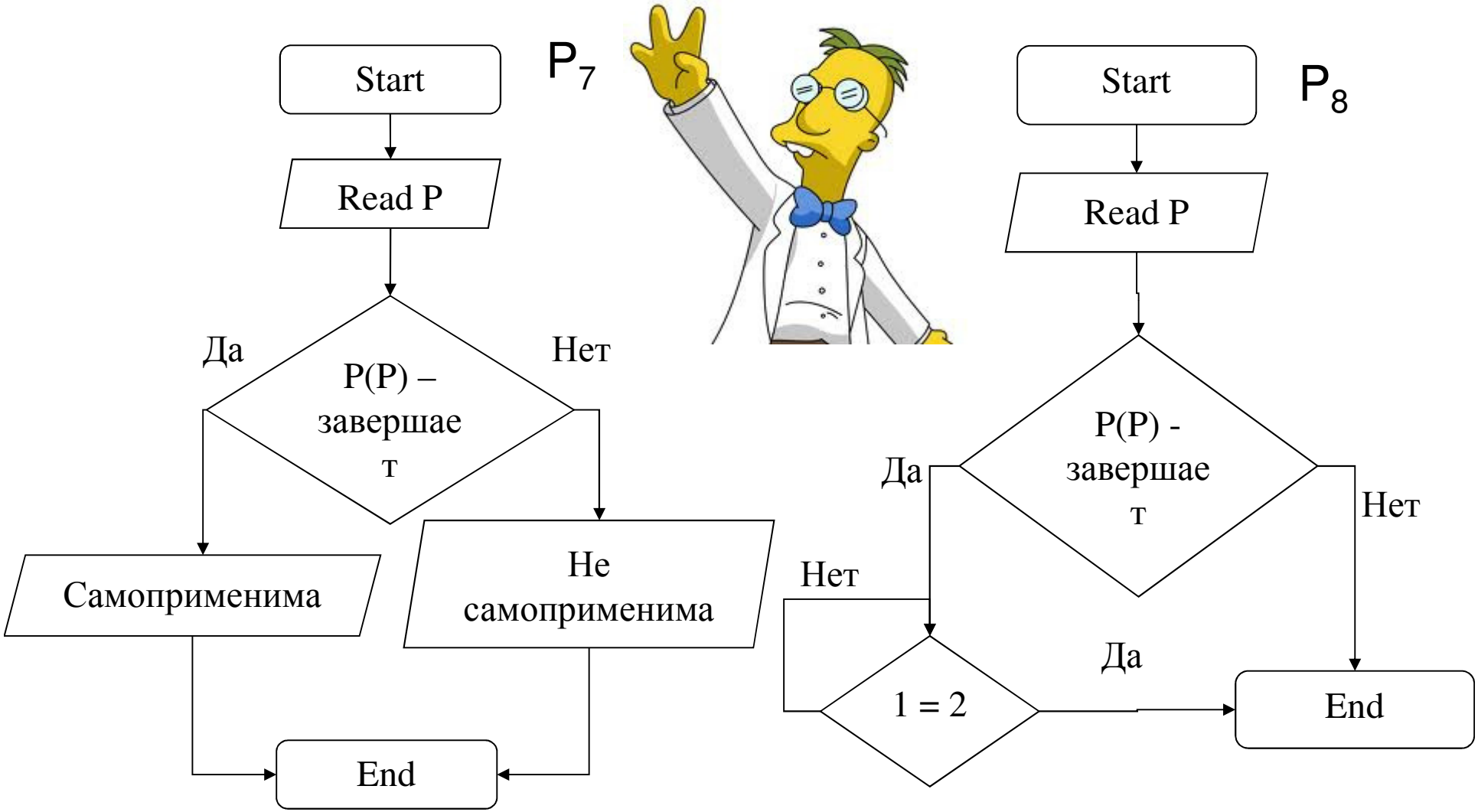
(от противного): Предположим, что существует программа P_7.



P_7

Доказательство леммы

(от противного): Предположим, что существует программа P_7. Тогда переделаем в ее программу P_8



Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.
Тогда «$P_8(P_8)$ – завершает ?» - Да.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.
Тогда «$P_8(P_8)$ – завершает ?» - Да,
Тогда P_8 –самоприменима.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.
Тогда «$P_8(P_8)$ – завершает ?» - Да.
Тогда P_8 –самоприменима.
Противоречие.

Доказательство леммы

(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - Нет.
Тогда P_8 – не самоприменима.
Противоречие.

Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.
Тогда «$P_8(P_8)$ – завершает ?» - Да.
Тогда P_8 –самоприменима.
Противоречие.

Поэтому, P_8 - не существует.

Доказательство леммы

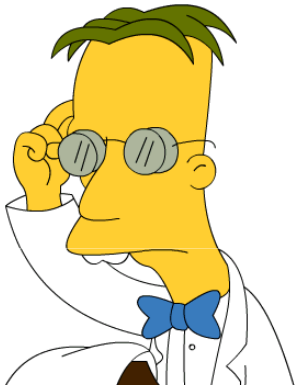
(от противного): Предположим, что существует программа P_7, тогда существует программа P_8

Программа P_8 – самоприменима или не самоприменима?

Пусть P_8 – самоприменима.
Тогда $P_8(P_8)$ – пришла в «End»
Тогда «$P_8(P_8)$ – завершает ?» - нет.
Тогда P_8 – не самоприменима.
Противоречие.

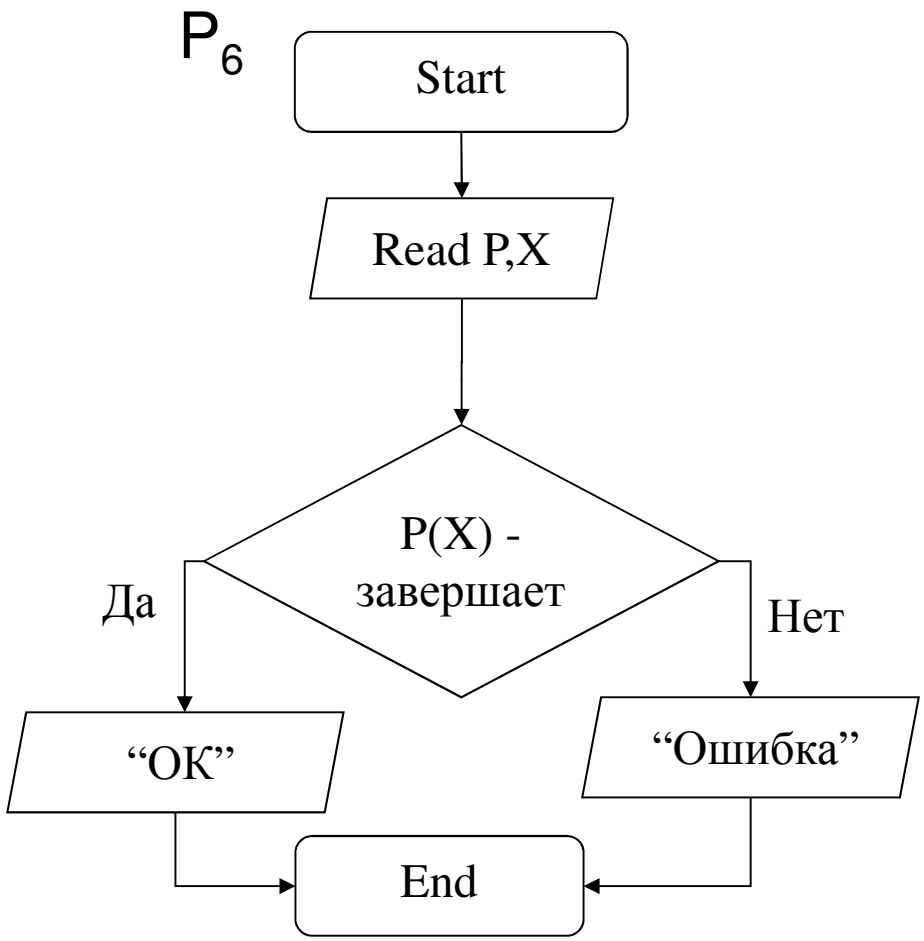
Пусть P_8 – не самоприменима,
Тогда $P_8(P_8)$ – зависла.
Тогда «$P_8(P_8)$ – завершает ?» - да,
тогда P_8 –самоприменима.
Противоречие.

Поэтому, P_8 - не существует.
Следовательно P_7 – не существует.



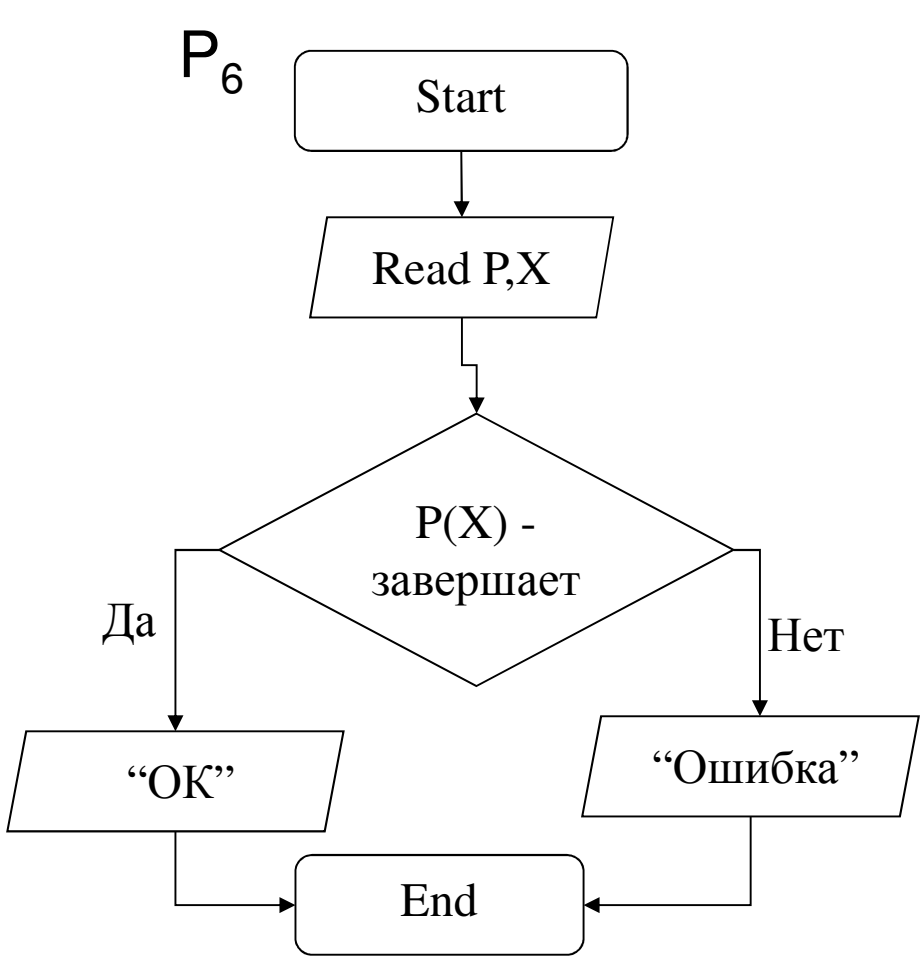
Доказательство теоремы

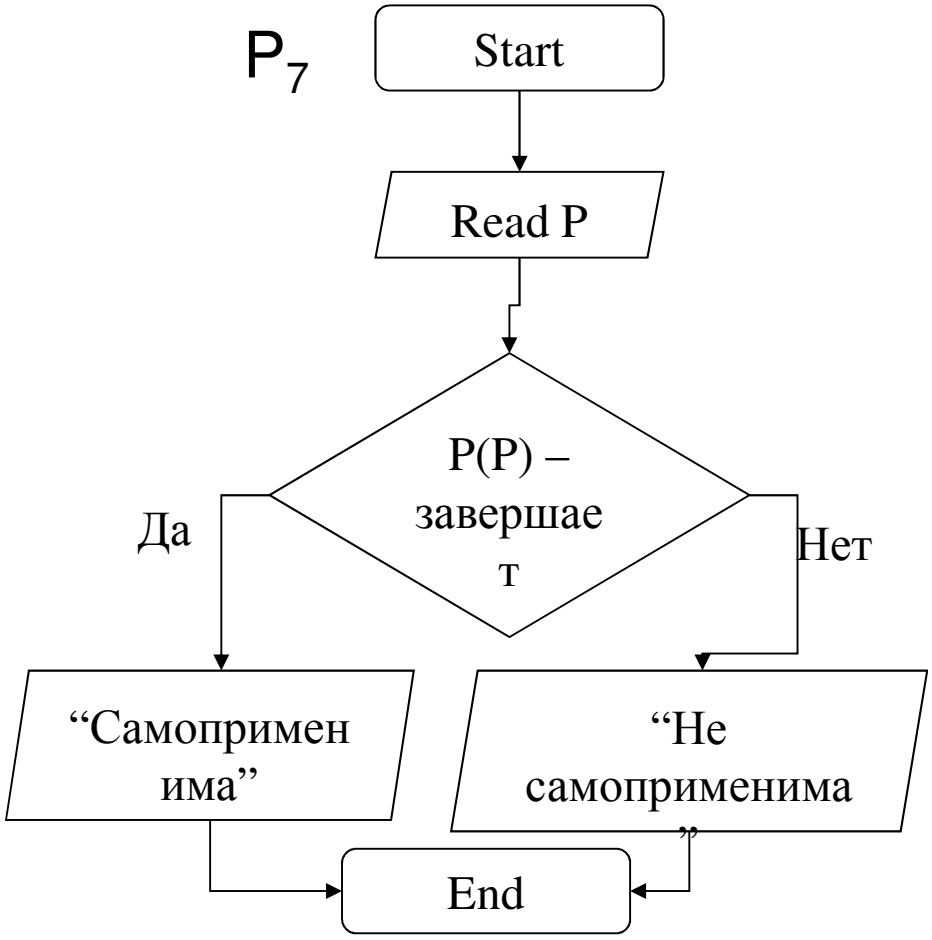
(от противного): Предположим что, существует P_6.



Доказательство теоремы

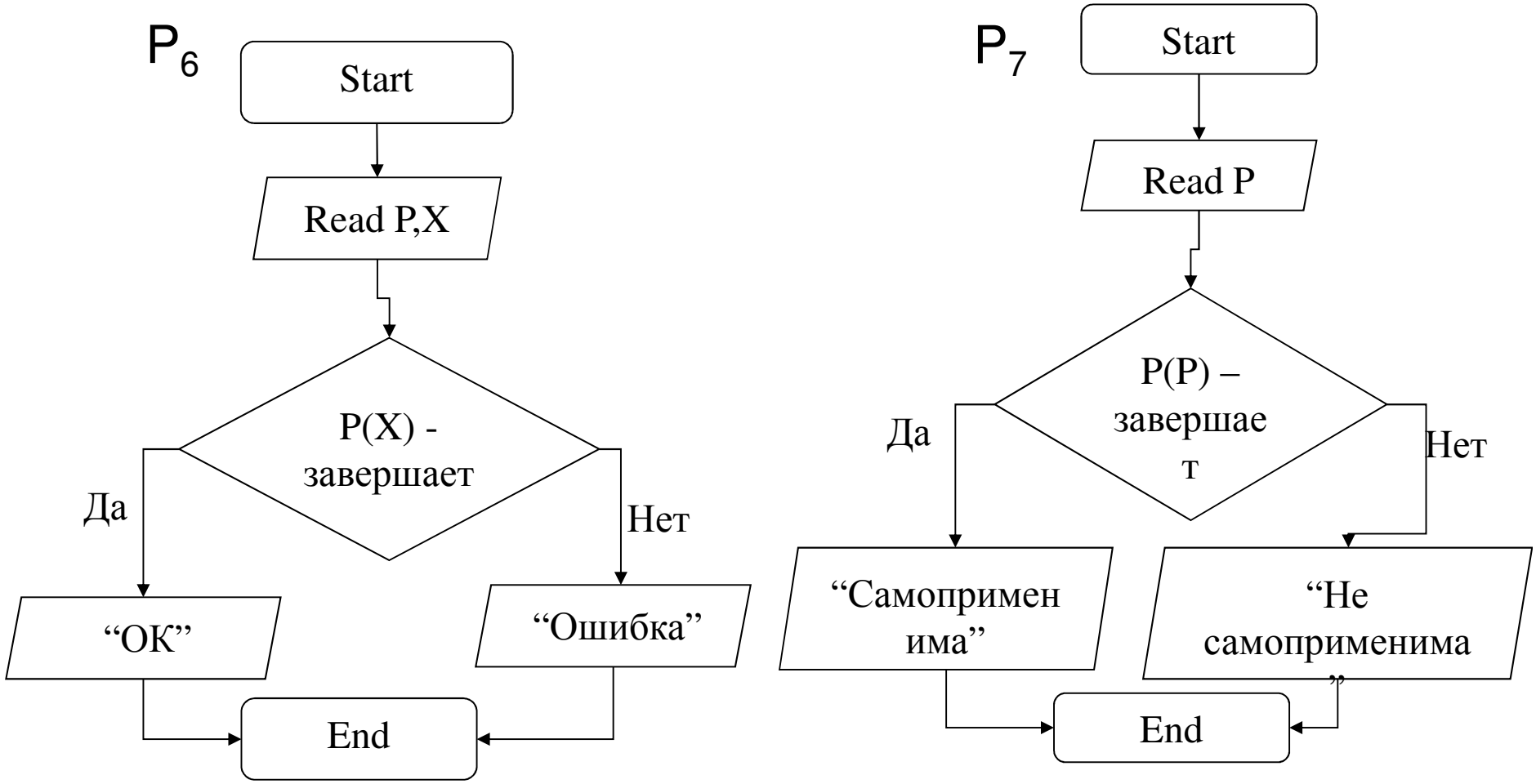
(от противного): Предположим что, существует P_6.
Тогда существует P_7, т.к. $P_7(T) = P_6(T,T)$





Доказательство теоремы

(от противного): Предположим что, существует P_6.
Тогда существует P_7, т.к. $P_7(T) = P_6(T,T)$



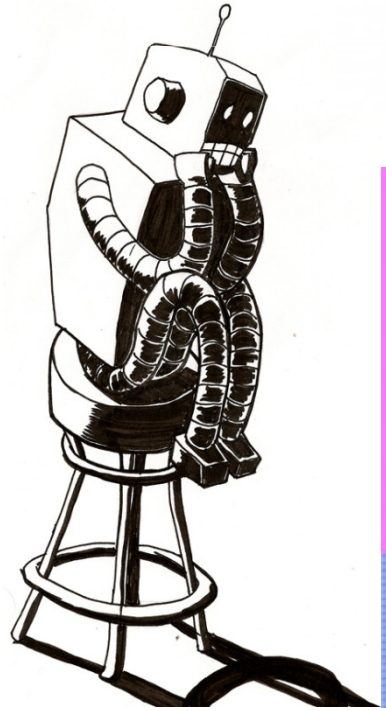
Но, P_7 - не существует, следовательно, P_6 – не существует

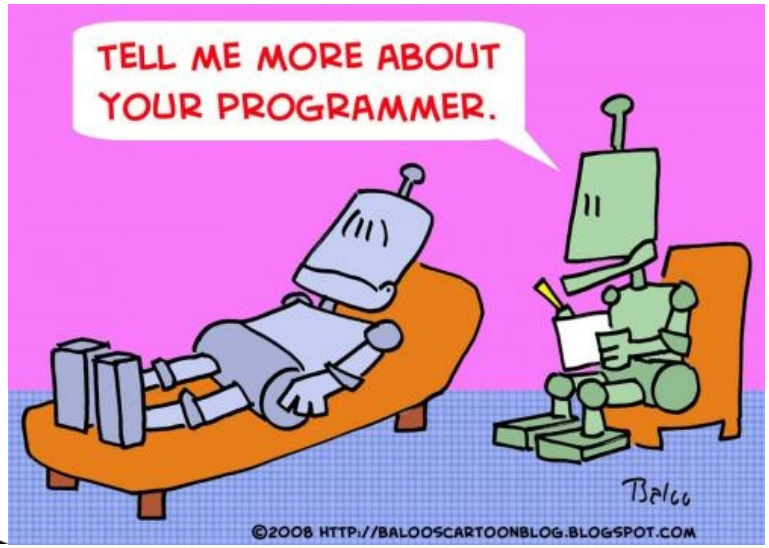
Искусственный интеллект?

Главная идея лекции:

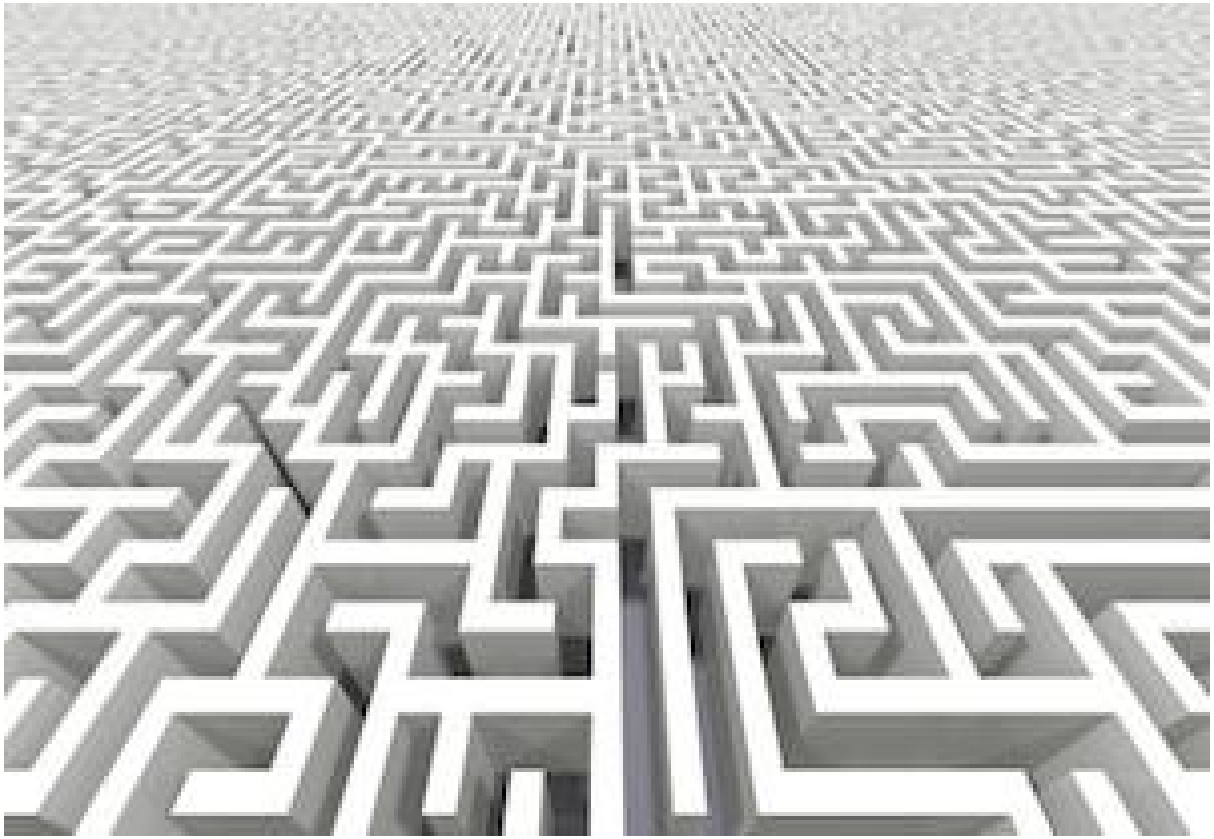
Робот (компьютер, программа) не может быть программистом.

Программистом может быть только человек (или иное живое существо).

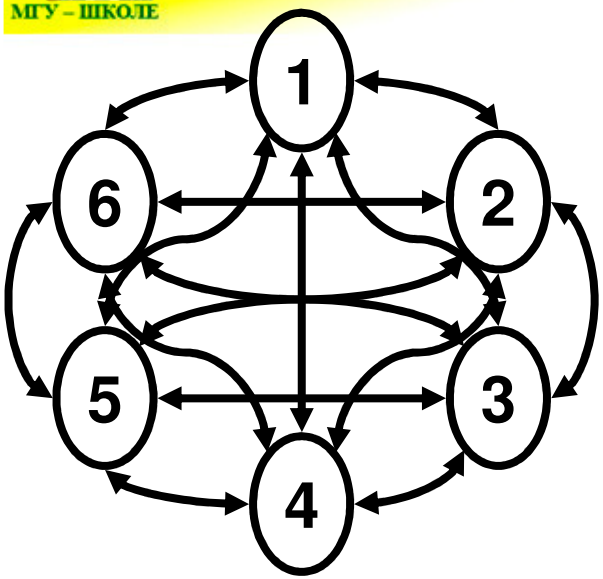




Часть 3
Труднорешаемые проблемы

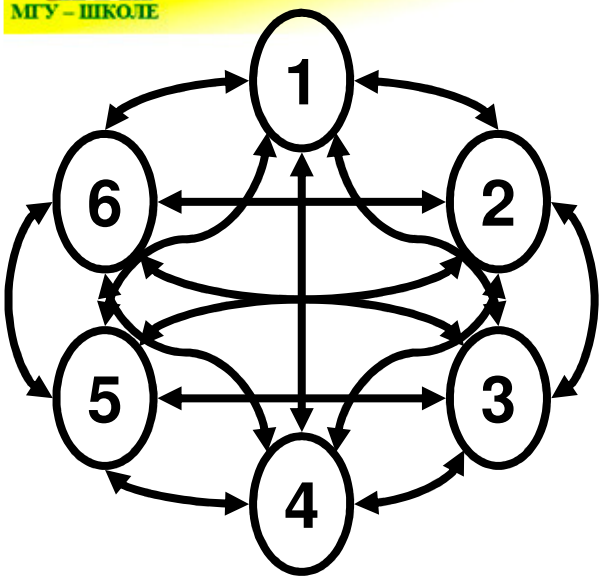


Эффективность работы



Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи.

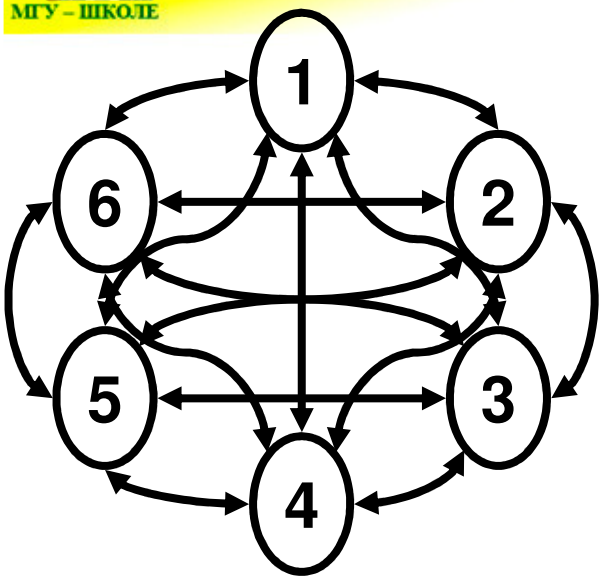
Эффективность работы



	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей.

Эффективность работы

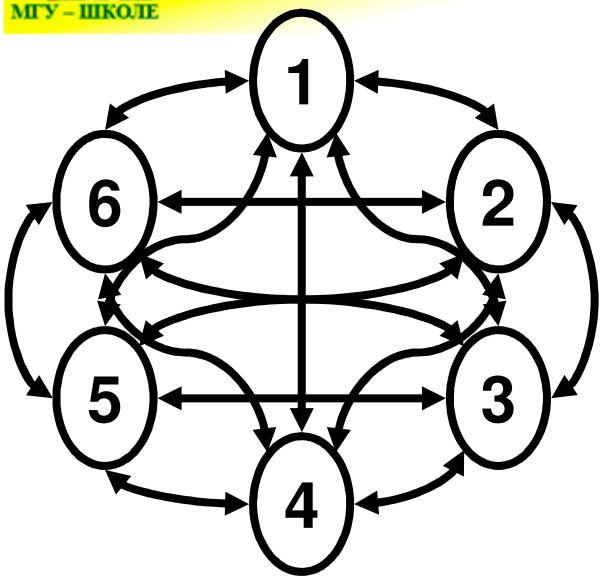


F({1,2,3}) = 1 - 2 + 5 = 4

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей.

Эффективность работы



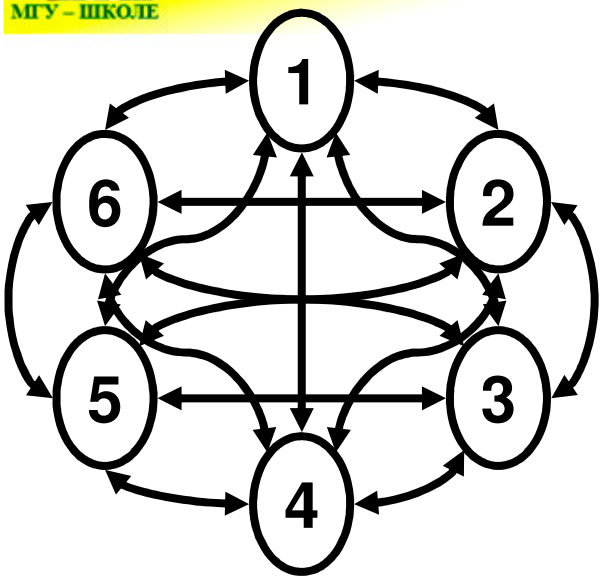
F({1,2,3}) = 1 - 2 + 5 = 4

F({3,5,6}) = 0 - 2 -1 = - 3

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей.

Эффективность работы

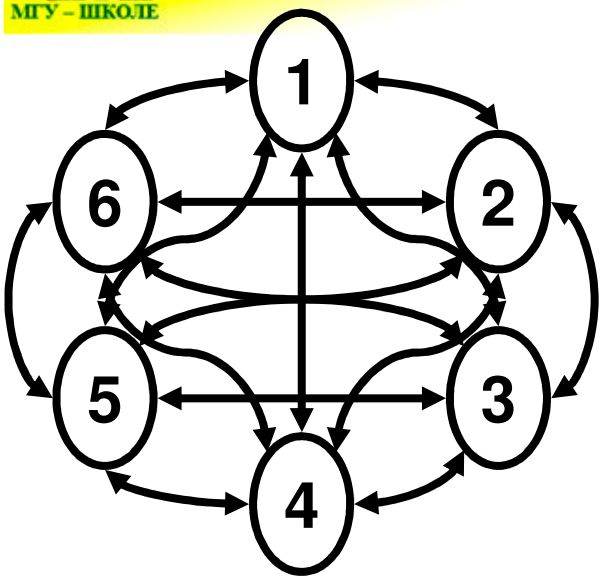


	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

F({1,2,3}) = 1 - 2 + 5 = 4
F({3,5,6}) = 0 - 2 -1 = - 3
F({2,4,5}) = 2 + 1 + 4 = 7

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей.

Эффективность работы



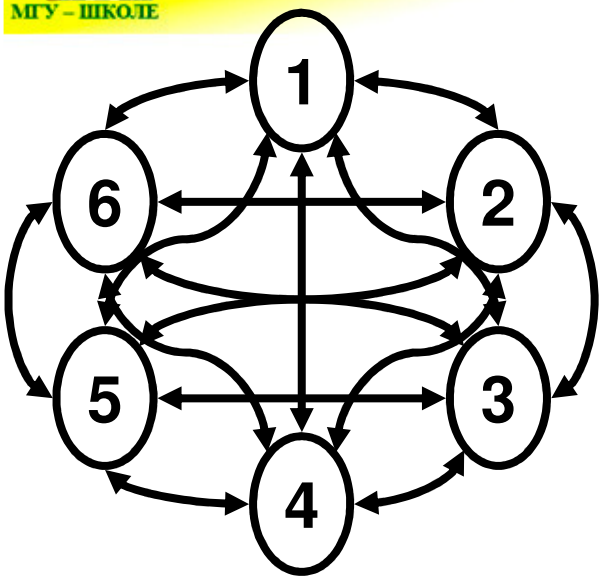
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

F({1,2,3}) = 1 - 2 + 5 = 4
F({3,5,6}) = 0 - 2 -1 = - 3
F({2,4,5}) = 2 + 1 + 4 = 7

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей.

Необходимо выбрать группу, которая была бы максимально эффективна. Как это сделать?

Эффективность работы

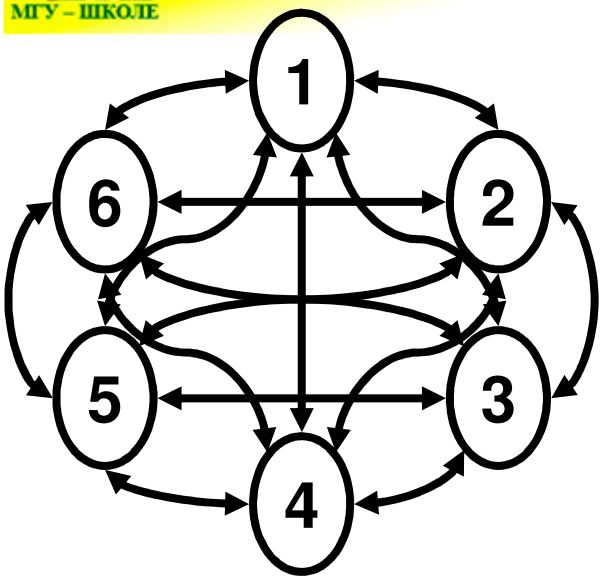


	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

F({1,2,3}) = 1 - 2 + 5 = 4
F({3,5,6}) = 0 - 2 -1 = - 3
F({2,4,5}) = 2 + 1 + 4 = 7

Есть коллектив из 6 человек, которые друг с другом взаимодействуют при решении некоторой задачи. Эффективность описывается матрицей. Необходимо выбрать группу, которая была бы максимально эффективна. Как это сделать? Перебрать всевозможные группы. **Всего 2^6 -1 = 63 варианта**

Эффективность работы



	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

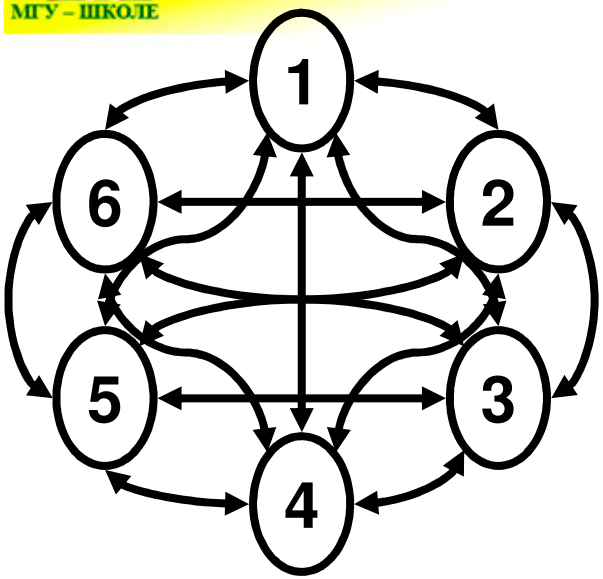
F({1,2,3}) = 1 - 2 + 5 = 4

F({3,5,6}) = 0 - 2 -1 = - 3

F({2,4,5}) = 2 + 1 + 4 = 7

Если n = 1000, то 2^n-1=

Эффективность работы

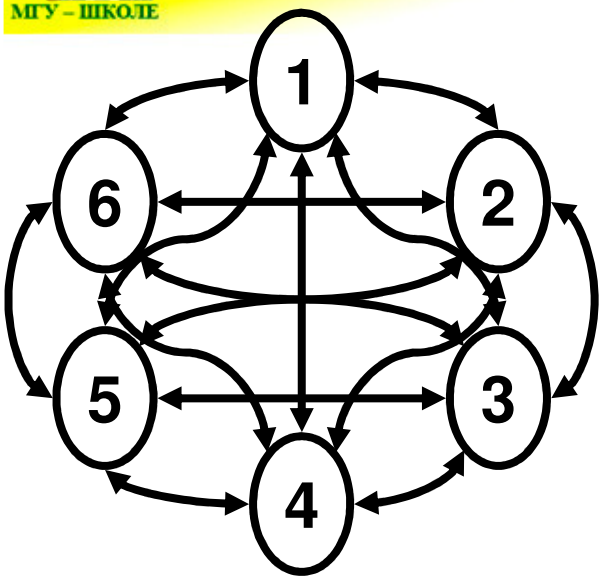


	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

F({1,2,3}) = 1 - 2 + 5 = 4
F({3,5,6}) = 0 - 2 -1 = - 3
F({2,4,5}) = 2 + 1 + 4 = 7

Если n = 1000, то 2^n-1= 10715086071862673209484250490600
018105614048117055336074437503883703510511249361224 9
319837881569585812759467291755314682518714528569231 4
043598457757469857480393456777482423098542107460506 2
371141877954182153046474983581941267398767559165543 9
460770629145711964776865421676604298316526243868372 0
5668069375 вариантов

Эффективность работы



	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

F({1,2,3}) = 1 - 2 + 5 = 4

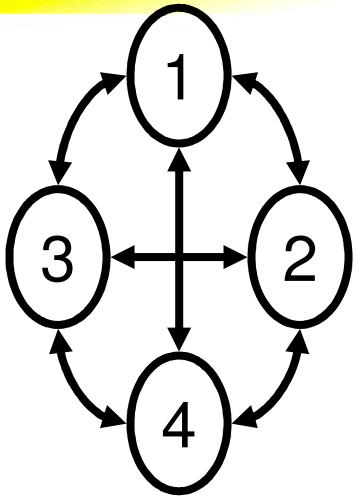
F({3,5,6}) = 0 - 2 -1 = - 3

F({2,4,5}) = 2 + 1 + 4 = 7

Если n = 1000, то 2^n-1= 10715086071862673209484250490600
018105614048117055336074437503883703510511249361…..

Есть ли другие варианты поиска оптимальной группы?

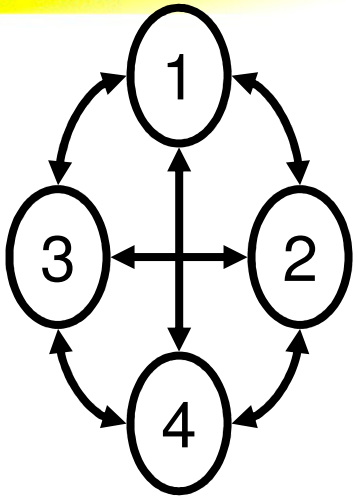
Жадный алгоритм



	1	**2**	**3**	**4**
1	0	5	4	3
2	5	0	-5	-5
3	4	-5	0	-1
4	3	-5	-1	0

«Жадный» алгоритм – пытаемся сразу набрать как можно больше.

Жадный алгоритм

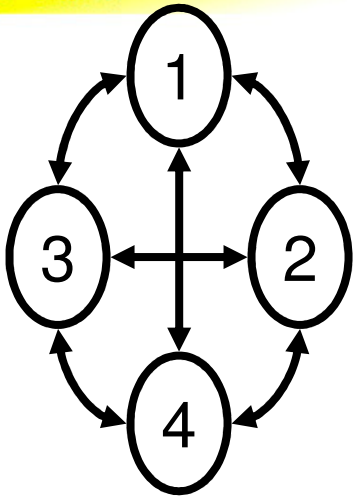


	1	**2**	**3**	**4**
1	0	5	4	3
2	5	0	-5	-5
3	4	-5	0	-1
4	3	-5	-1	0

«Жадный» алгоритм – пытаемся сразу набрать как можно больше.

F({1,2}) = 5

Жадный алгоритм



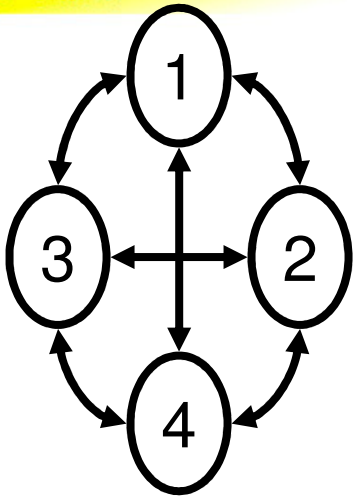
	1	**2**	**3**	**4**
1	0	5	4	3
2	5	0	-5	-5
3	4	-5	0	-1
4	3	-5	-1	0

«Жадный» алгоритм – пытаемся сразу набрать как можно больше.

F({1,2}) = 5

F({1,2,3}) = 5 + 4 - 5= 4

Жадный алгоритм



	1	**2**	**3**	**4**
1	0	5	4	3
2	5	0	-5	-5
3	4	-5	0	-1
4	3	-5	-1	0

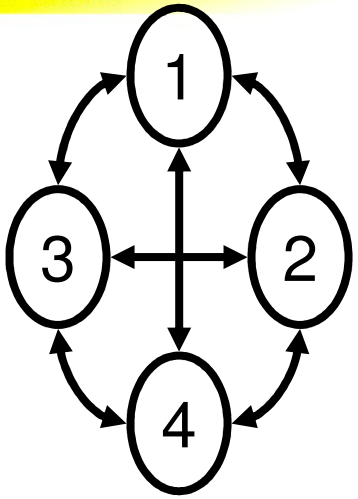
«Жадный» алгоритм – пытаемся сразу набрать как можно больше.

F({1,2}) = 5

F({1,2,3}) = 5 + 4 - 5= 4

F({1,2,3,4}) = 5 + 4 + 3 - 5 - 5 - 1 = 1

Жадный алгоритм



	1	**2**	**3**	**4**
1	0	5	4	3
2	5	0	-5	-5
3	4	-5	0	-1
4	3	-5	-1	0

«Жадный» алгоритм – пытаемся сразу набрать как можно больше.

F({1,2}) = 5

F({1,2,3}) = 5 + 4 - 5= 4

F({1,2,3,4}) = 5 + 4 + 3 - 5 - 5 - 1 = 1

но: F({1,3,4}) = 4 + 3 - 1 = 6

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

Возможны два варианта:

- Алгоритм есть, но его пока не придумали
- Эффективного алгоритма не существует

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

Возможны два варианта:

- Алгоритм есть, но его пока не придумали
- Эффективного алгоритма не существует

Как Вы думаете, какой вариант правильный?

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

Возможны два варианта:

- Алгоритм есть, но его пока не придумали
- Эффективного алгоритма не существует

Как Вы думаете, какой вариант правильный?

Какой вариант правильный – неизвестно ...

МГУ – ШКОЛЕ

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

Возможны два варианта:

- Алгоритм есть, но его пока не придумали
- Эффективного алгоритма не существует

Как Вы думаете, какой вариант правильный?

Какой вариант правильный – неизвестно ...

Это: P = NP проблема

P = NP проблема

Если ли другой алгоритм решения этой задачи кроме полного перебора?

Возможны два варианта:

- Алгоритм есть, но его пока не придумали
- Эффективного алгоритма не существует

Как Вы думаете, какой вариант правильный?

Какой вариант правильный – неизвестно ...

Это: P = NP проблема

Но вся криптография «держится» на этом!

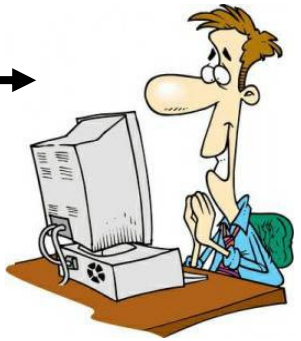
Переборные задачи в криптографии

Задача: Надо отправить «да» или «нет» от одного пользователя к другому, но сообщение надо зашифровать

Переборные задачи в криптографии

Задача: Надо отправить «да» или «нет» от одного пользователя к другому, но сообщение надо зашифровать

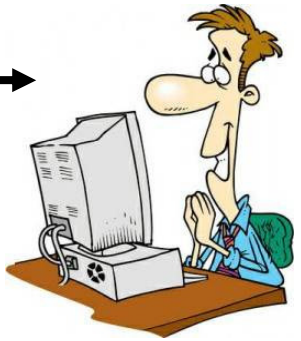
Перехват сообщения



Переборные задачи в криптографии

Задача: Надо отправить «да» или «нет» от одного пользователя к другому, но сообщение надо зашифровать

Перехват сообщения



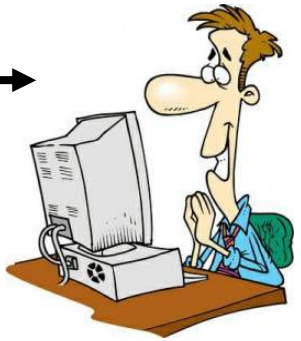
Ключом (паролем) K является множество чисел, например 2,4,5.

Его знают только отправитель и получатель.

Переборные задачи в криптографии

Задача: Надо отправить «да» или «нет» от одного пользователя к другому, но сообщение надо зашифровать

Перехват сообщения



Ключом (паролем) K является множество чисел, например 2,4,5.

Его знают только отправитель и получатель.
Генерируем матрицу M,
такую что F(K) – максимально.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Переборные задачи в криптографии

Задача: Надо отправить «да» или «нет» от одного пользователя к другому, но сообщение надо зашифровать

Перехват сообщения

Ключом (паролем) K является множество чисел, например 2,4,5.

Его знают только отправитель и получатель.
Генерируем матрицу M,
такую что F(K) – максимально.

Посылаем по каналу связи M и число R:
Если «Да», то F(K) = R
Если «Нет», то F(K) < R

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Переборные задачи в криптографии

Перехват сообщения

Ключом (паролем) K является множество чисел, например 2,4,5.

Его знают только отправитель и получатель.
Генерируем матрицу M,
такую что F(K) – максимально.

Посылаем по каналу связи M и число R:
Если «Да», то F(K) = R
Если «Нет», то F(K) < R

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Получатель легко поймет, что было послано, т.к. знает M, K и R.
Он вычисляет F(K) при помощи M и сравнивает с R.

Переборные задачи в криптографии

Перехват сообщения

Ключом (паролем) K является множество чисел, например 2,4,5.

Его знают только отправитель и получатель.
Генерируем матрицу M,
такую что F(K) – максимально.

Посылаем по каналу связи M и число R:
Если «Да», то F(K) = R
Если «Нет», то F(K) < R

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Получатель легко поймет, что было послано, т.к. знает M, K и R.
Он вычисляет F(K) при помощи M и сравнивает с R.

А злоумышленнику придется перебирать варианты, чтобы найти K.

МГУ – ШКОЛЕ

Часть 3
Биологические алгоритмы

Откажемся от идеальности

В природе нет ничего идеального.
Любые живые существа совершают ошибки.

Откажемся от идеальности

В природе нет ничего идеального.
Любые живые существа совершают ошибки.

Откажемся от идеальности

В природе нет ничего идеального.
Любые живые существа совершают ошибки.

Откажемся от идеальности

В природе нет ничего идеального.
Любые живые существа совершают ошибки.

Откажемся от идеальности

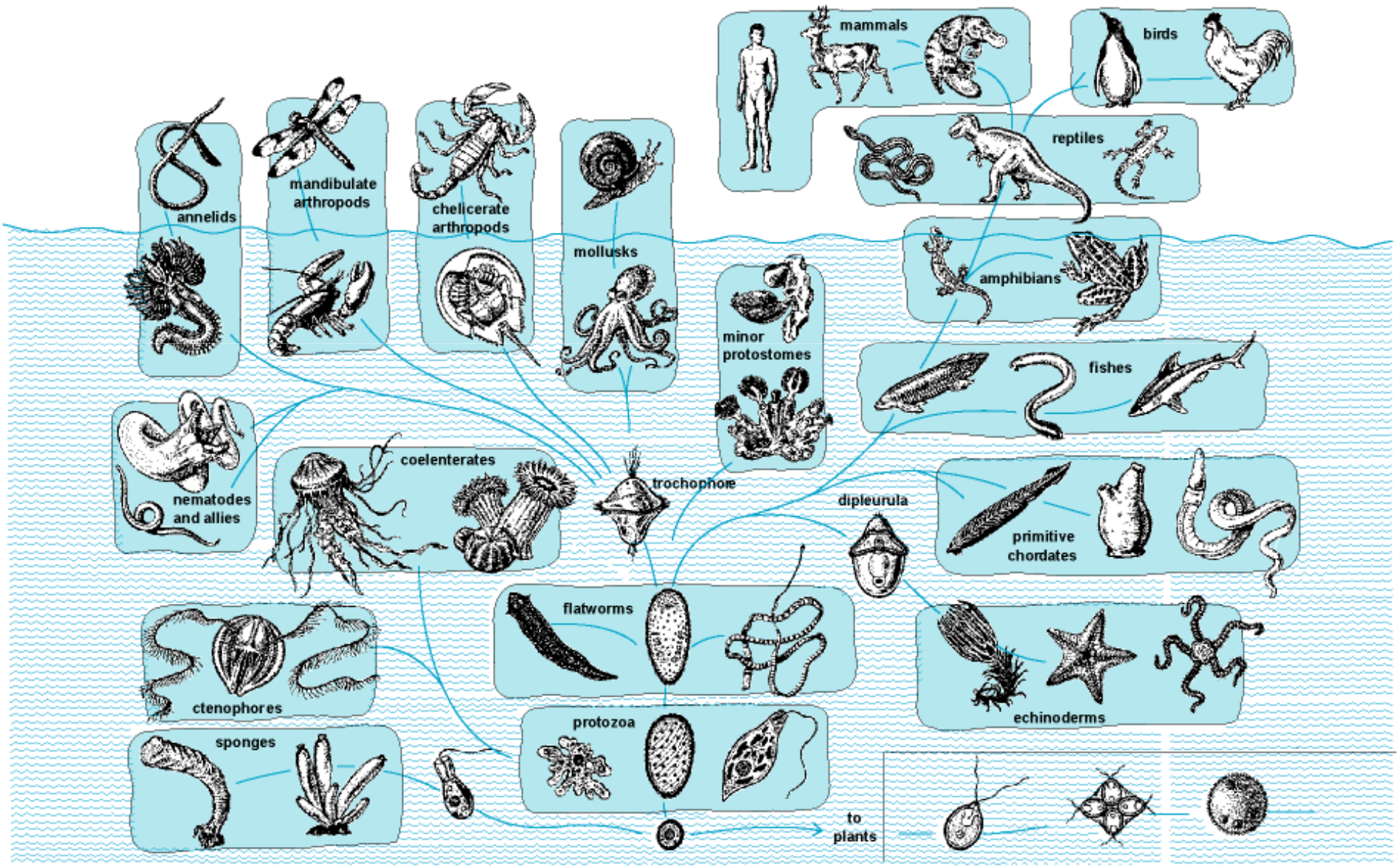
В природе нет ничего идеального.
Любые живые существа совершают ошибки.

Откажемся от поиска самой лучшей группы в нашей задаче.

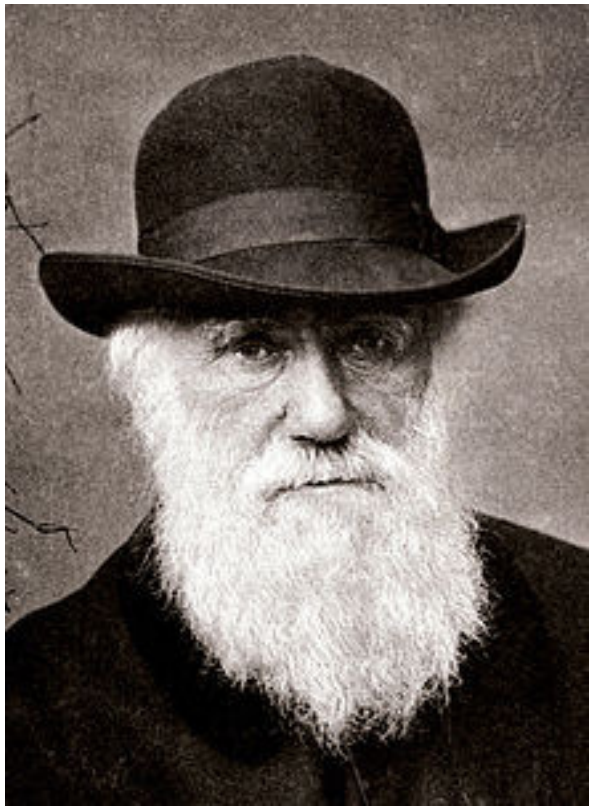
Будем искать более-менее хорошую группу в нашей задаче.

МГУ – ШКОЛЕ

Идея эволюции



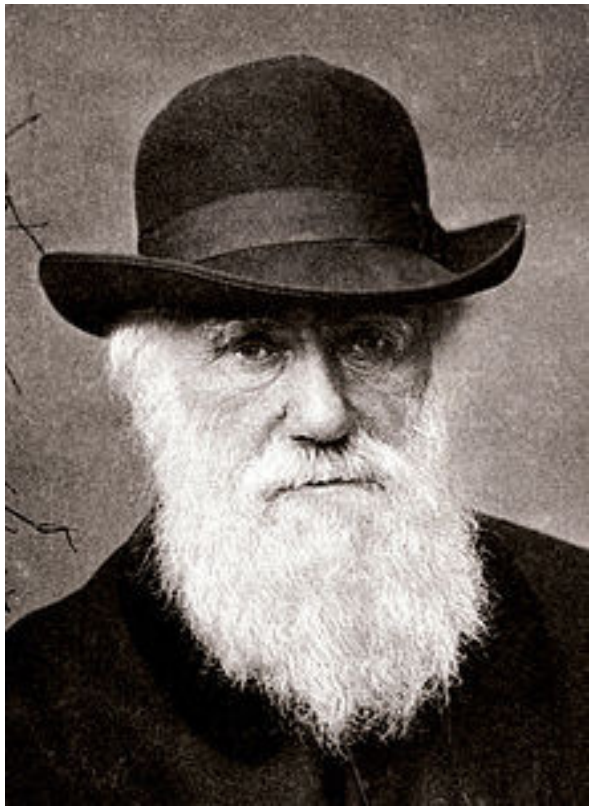
Идея эволюции



Чарлз Роберт Дарвин
английский натуралист
и путешественник

Идея эволюции

1. Во время рождения новых поколений происходят мутации, т.е. некоторые изменения в геноме животного.

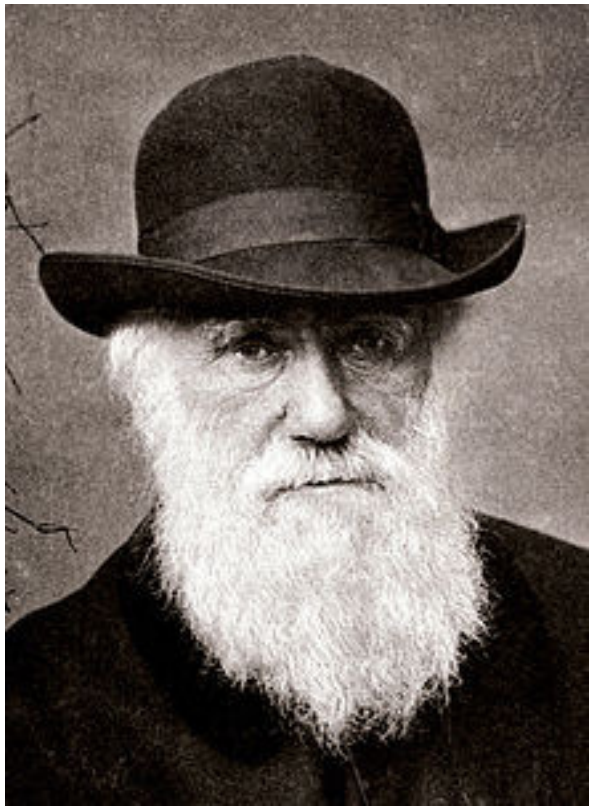


Чарлз Роберт Дарвин
английский натуралист
и путешественник

Идея эволюции

1. Во время рождения новых поколений происходят мутации, т.е. некоторые изменения в геноме животного.

2. Продолжают род те, кто наиболее приспособлен к окружающей среде.



Чарлз Роберт Дарвин
английский натуралист
и путешественник

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011 {2,3} – 011000

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011 {2,3} – 011000
Код K – это генетический код популяции.

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011 {2,3} – 011000
Код K – это генетический код популяции.

F(110100) = F({1,2,4}) = 1 + 2 - 3 = 0

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011 {2,3} – 011000
Код K – это генетический код популяции.

F(110100) = F({1,2,4}) = 1 + 2 - 3 = 0
F(110011) = F({1,2,5,6}) = 1 + 6 + 1 + 1 + 0 - 1 = 8

Генетические алгоритмы

Предположим, что точное решение нам не требуется, а необходимо более или менее хорошее решение.

	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу
Пример: {1,2,4} – 110100
{1,3,5} – 101010 {1,2,5,6} – 110011 {2,3} – 011000
Код K – это генетический код популяции.

F(110100) = F({1,2,4}) = 1 + 2 - 3 = 0
F(110011) = F({1,2,5,6}) = 1 + 6 + 1 + 1 + 0 - 1 = 8

Чем больше значение F(K), тем лучше популяция приспособлена к окружающей среде.

Генетические алгоритмы

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу

Код K – это генетический код популяции.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Чем больше значение F(K),
тем лучше популяция приспособлена к окружающей среде.

Генетические алгоритмы

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу

Код K – это генетический код популяции.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Чем больше значение F(K), тем лучше популяция приспособлена к окружающей среде.

Задача поиска группы, которая работает лучше всего эквивалентна задачи поиска популяции, которая лучше всего приспособлена к окружающей среде.

Генетические алгоритмы

Каждой группе сопоставим код длины 6. Код состоит из 0 и 1:
0 – элемент не входит группу
1 – элемент входит группу

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Код K – это генетический код популяции.

Чем больше значение F(K), тем лучше популяция приспособлена к окружающей среде.

Задача поиска группы, которая работает лучше всего эквивалентна задачи поиска популяции, которая лучше всего приспособлена к окружающей среде.

Мутация – случайное изменение 0 на 1 (или наоборот) в коде.

Генетические алгоритмы

111000

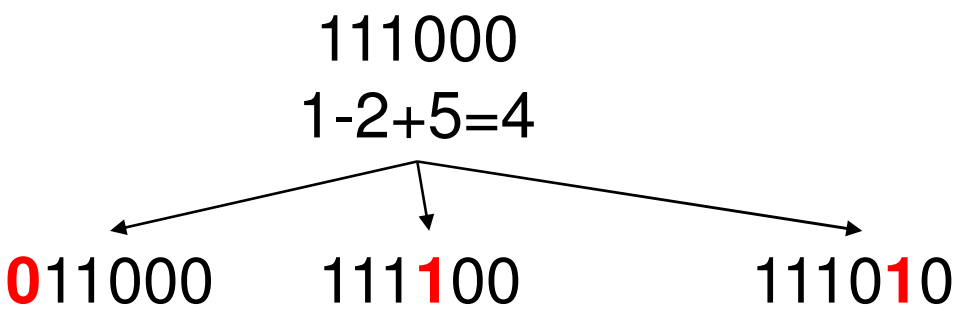
	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

111000
1-2+5=4

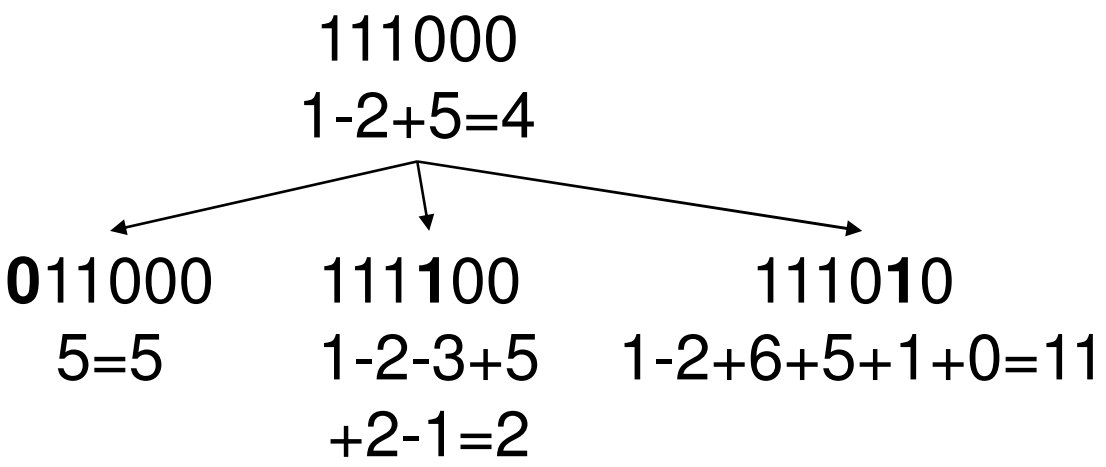
	1	**2**	**3**	**4**	**5**	**6**
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



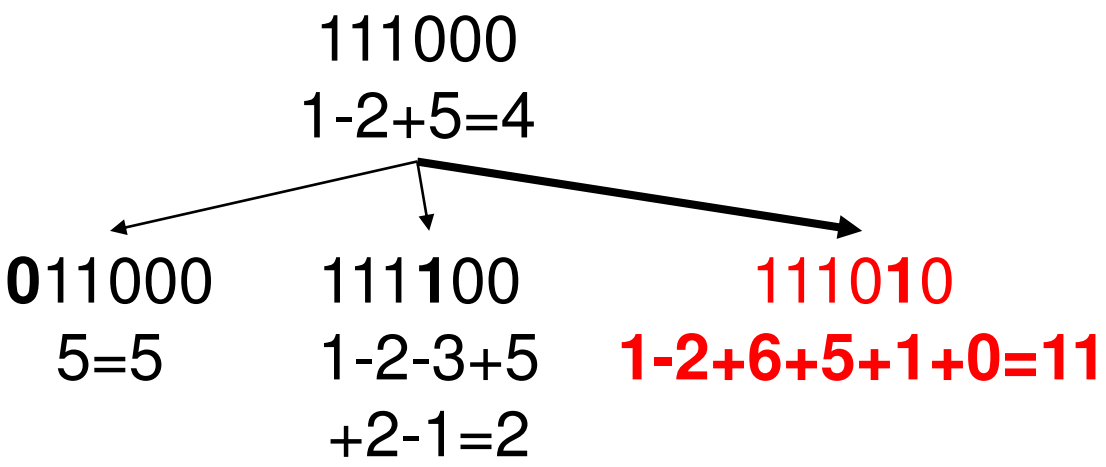
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



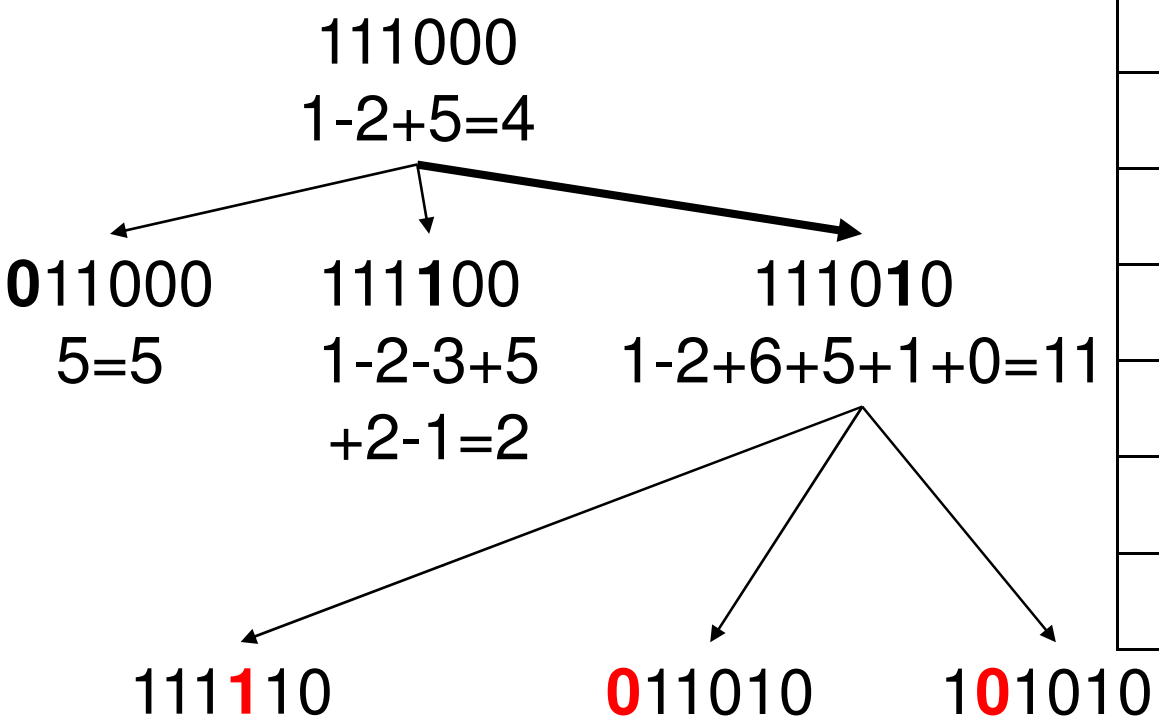
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



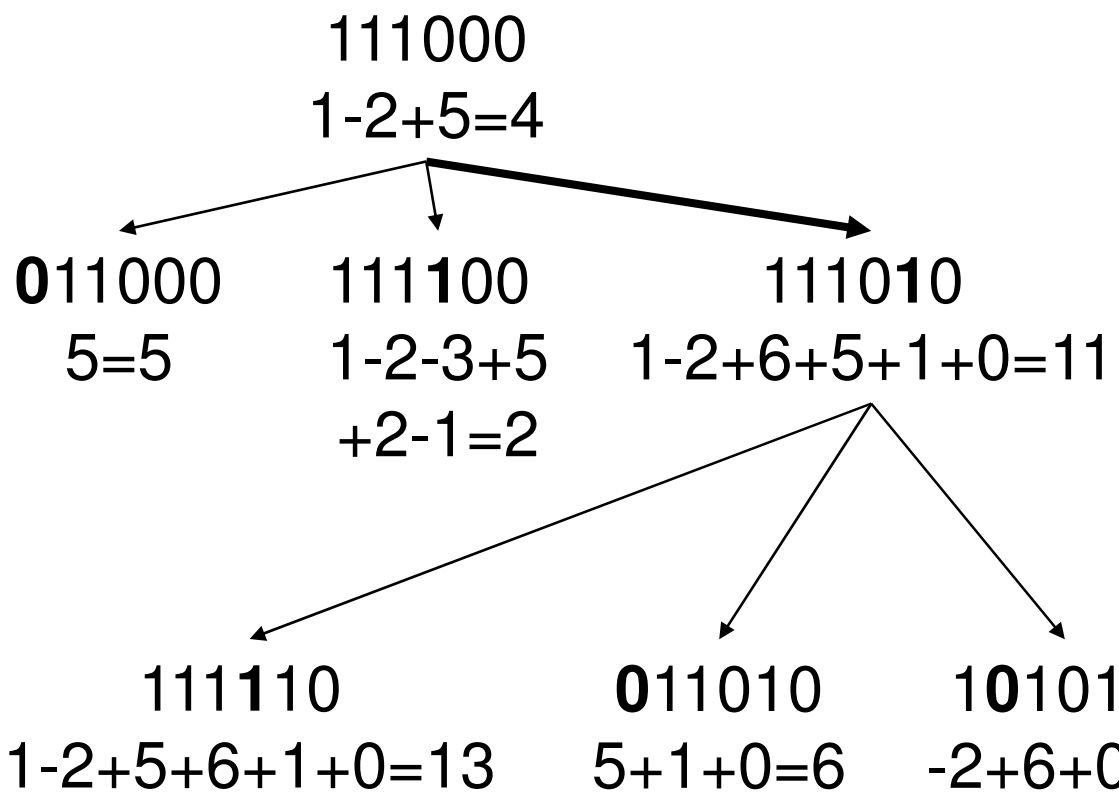
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



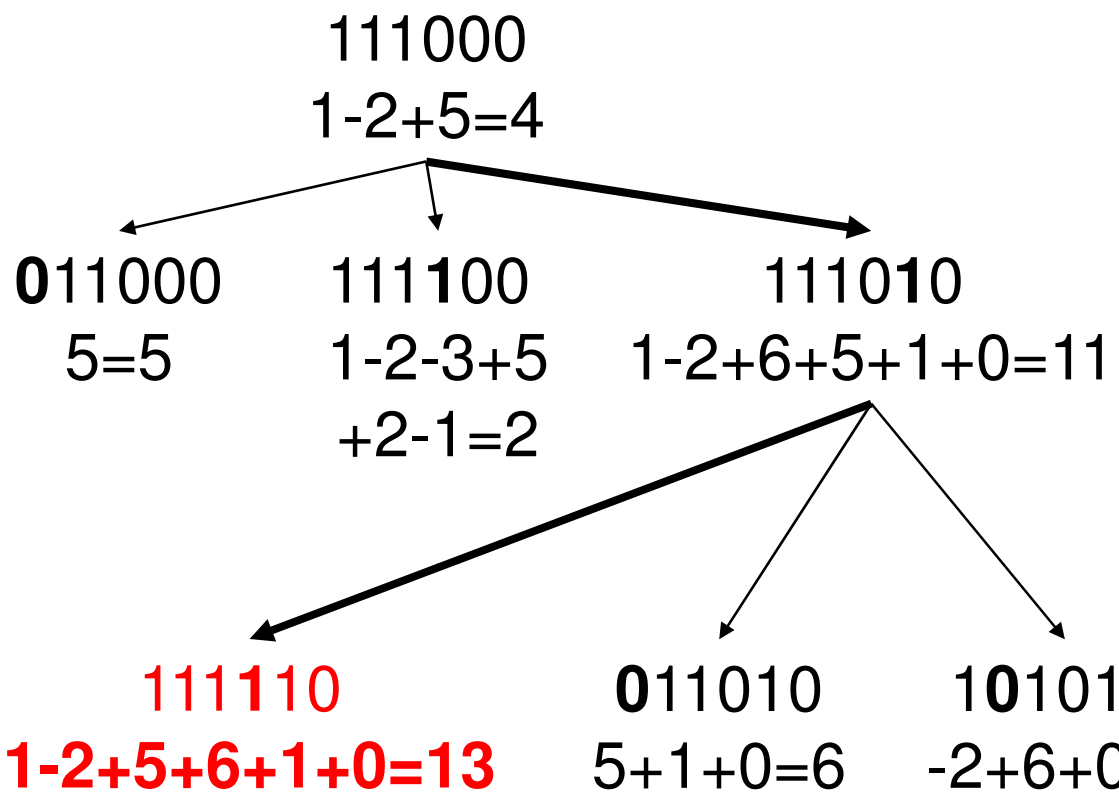
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



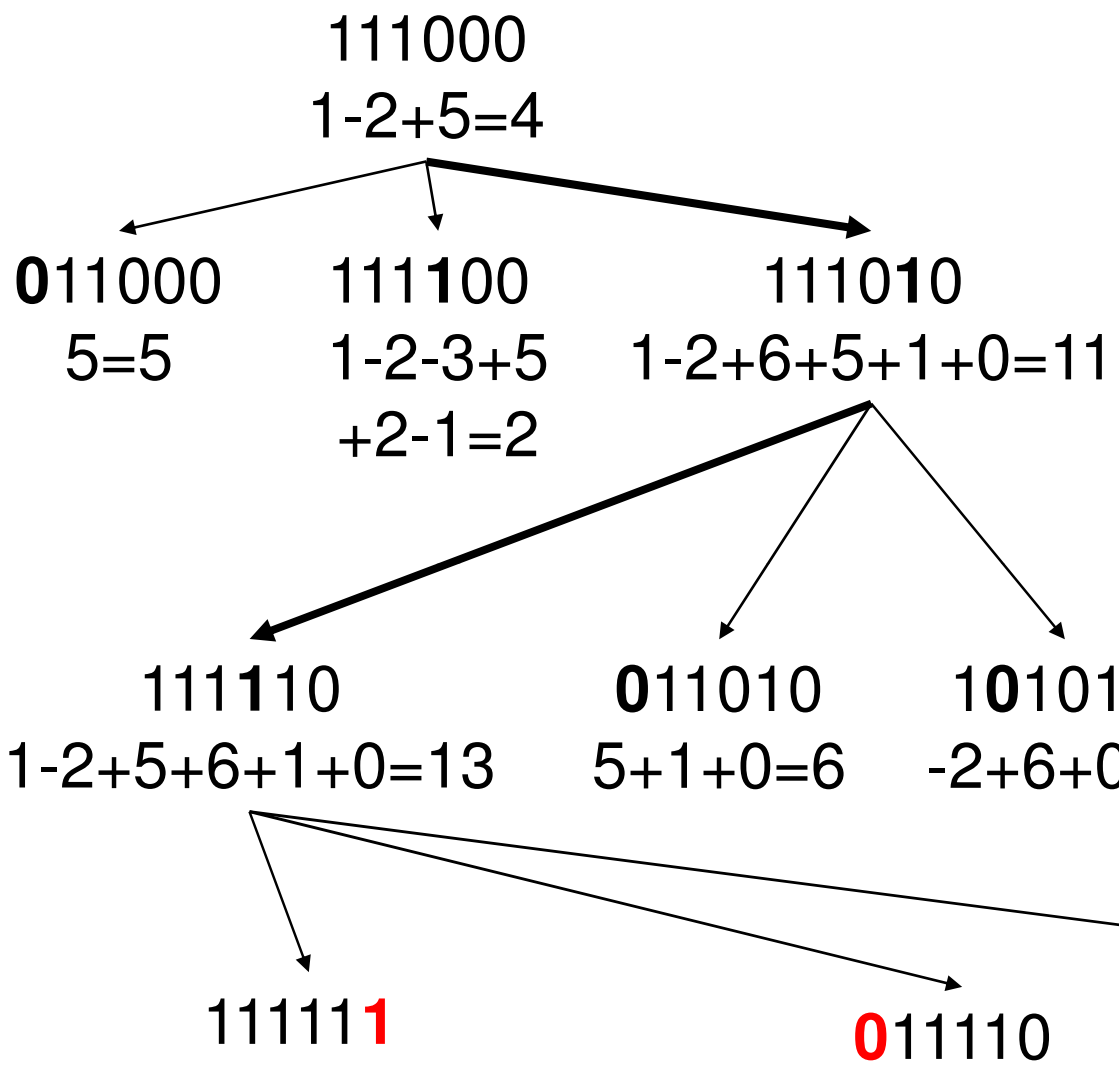
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



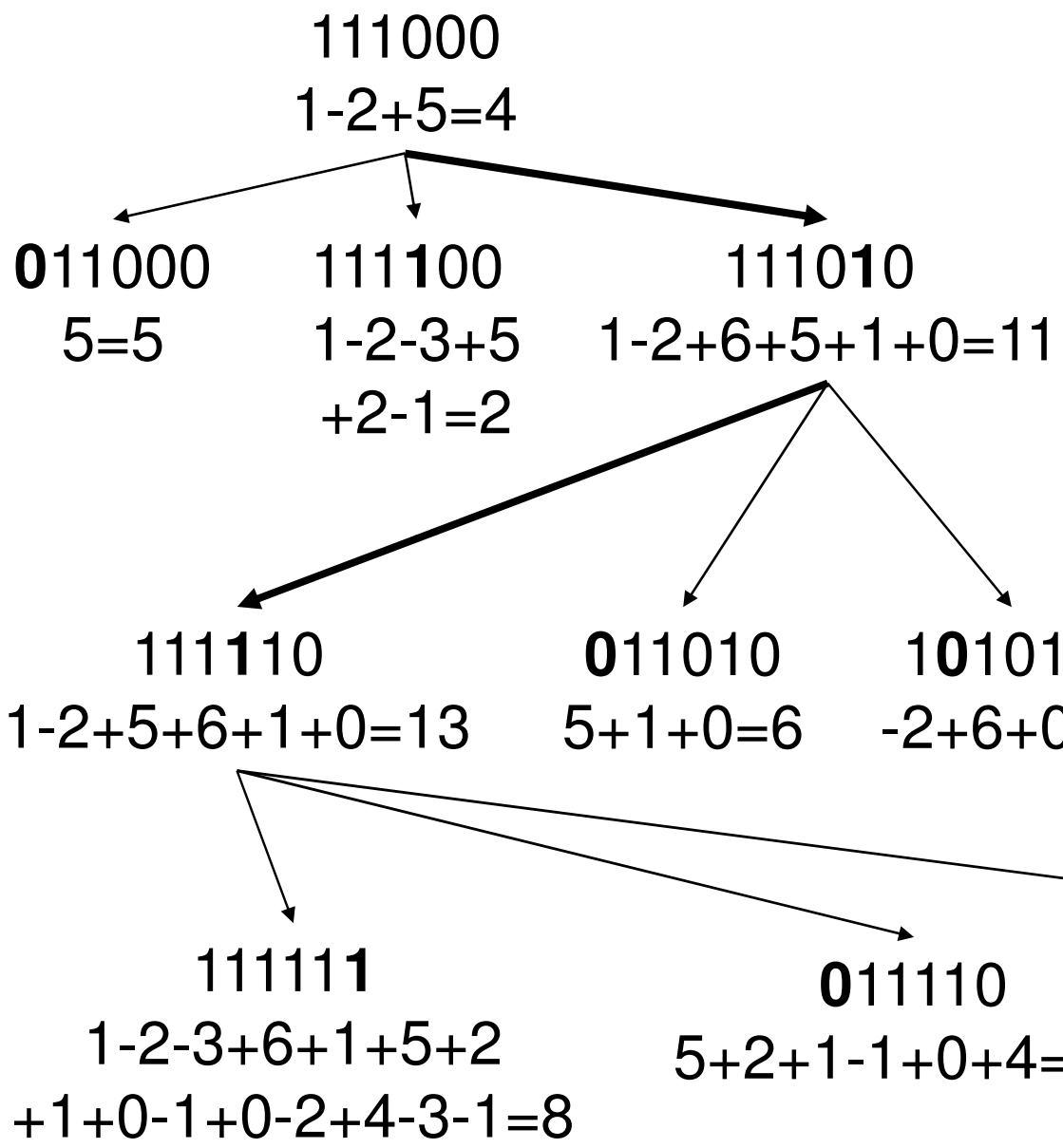
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы



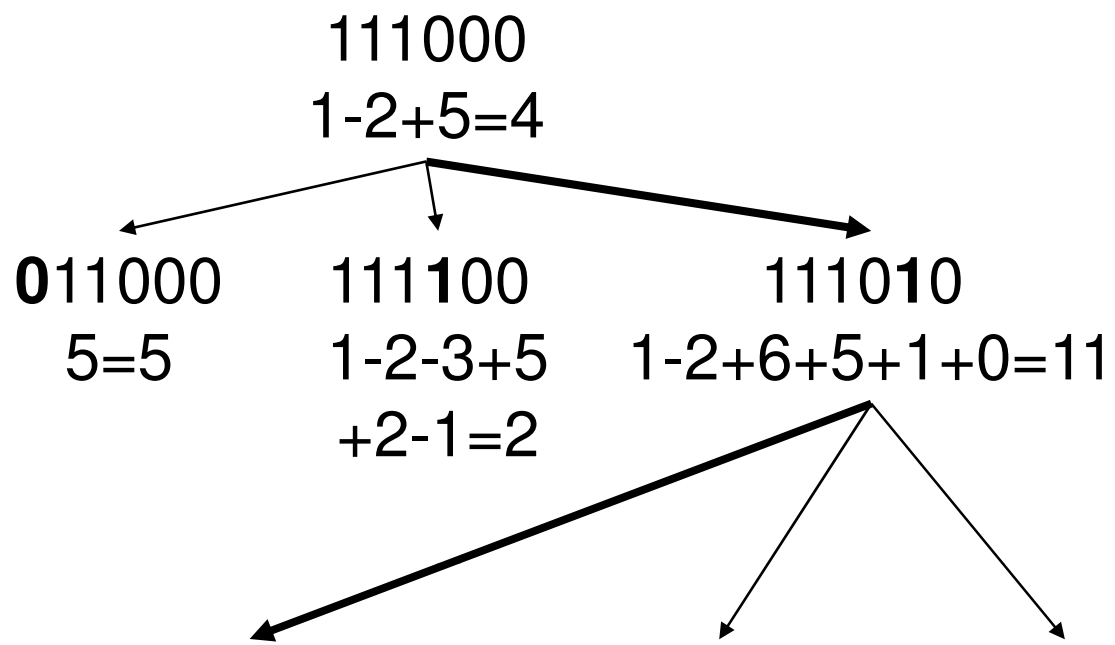
	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

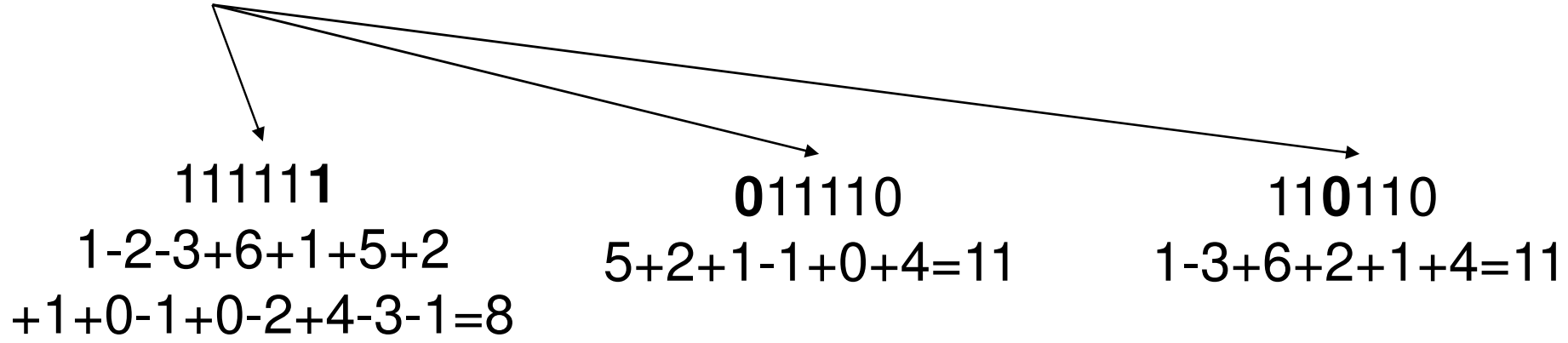


	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0

Генетические алгоритмы

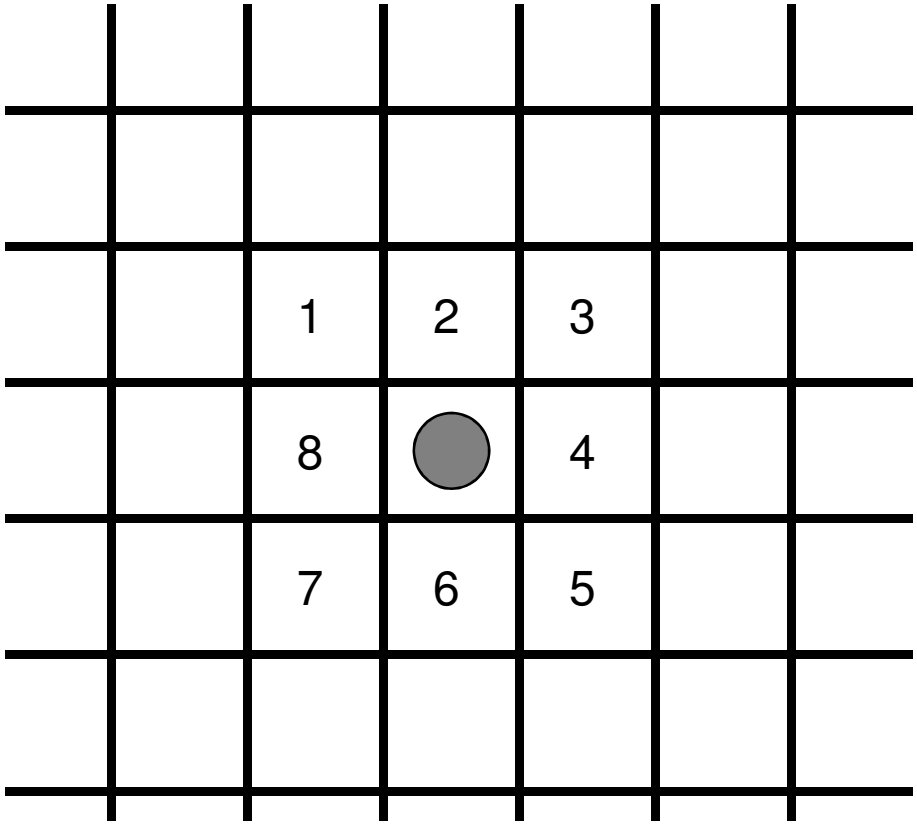


	1	2	3	4	5	6
1	0	1	-2	-3	6	1
2	1	0	5	2	1	0
3	-2	5	0	-1	0	-2
4	-3	2	-1	0	4	-3
5	6	1	0	4	0	-1
6	1	0	-2	-3	-1	0



Клеточные автоматы

Моделирование процесса жизни клеток.
Есть поле из ячеек, в каждой из которой может жить клетка.
Модель работает в дискретном времени (по тактам 1,2,3…)
Клетки рождаются или погибают по некоторым правилам,
в зависимости от того, есть или нет в соседних клетках.



У каждой клетки есть окрестность из 8 ячеек.

Клетка рождается, если в ее окрестности определенное число других клеток.

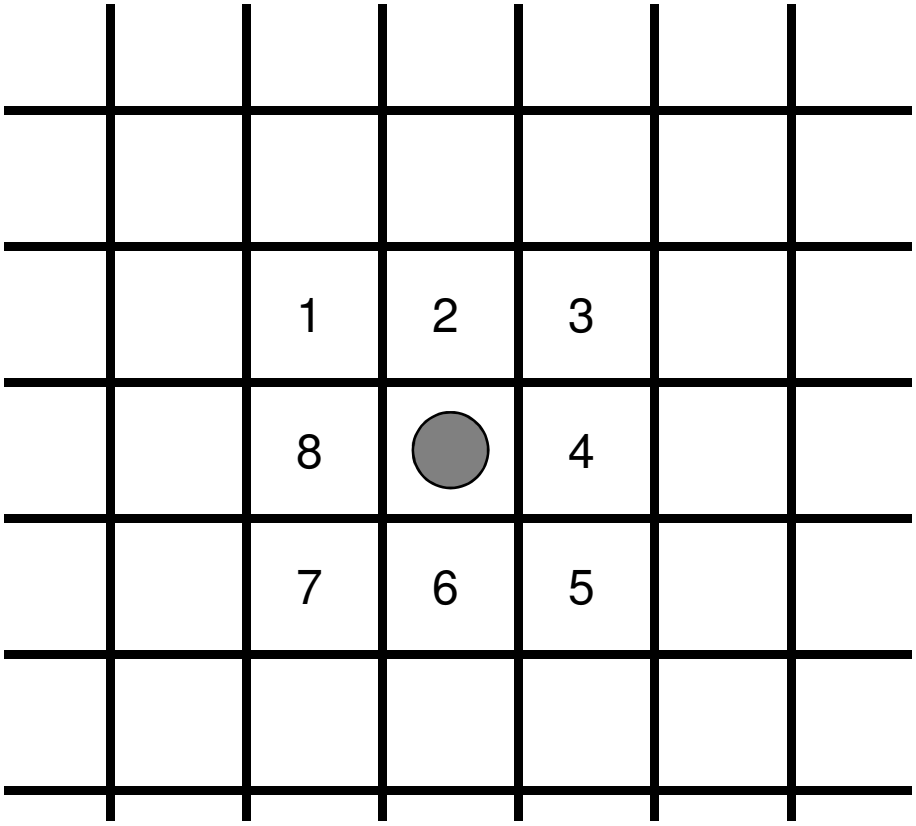
Клетка выживает, если в ее окрестности определенное число других клеток.

Клеточные автоматы

Пусть в ячейке **рождается** клетка, если в окрестности **1,2 или 3** другие клетки

Пусть в ячейке **выживает** клетка, если в окрестности **3,4 или 5** другие клетки

Это правило записывается **B123/S345**



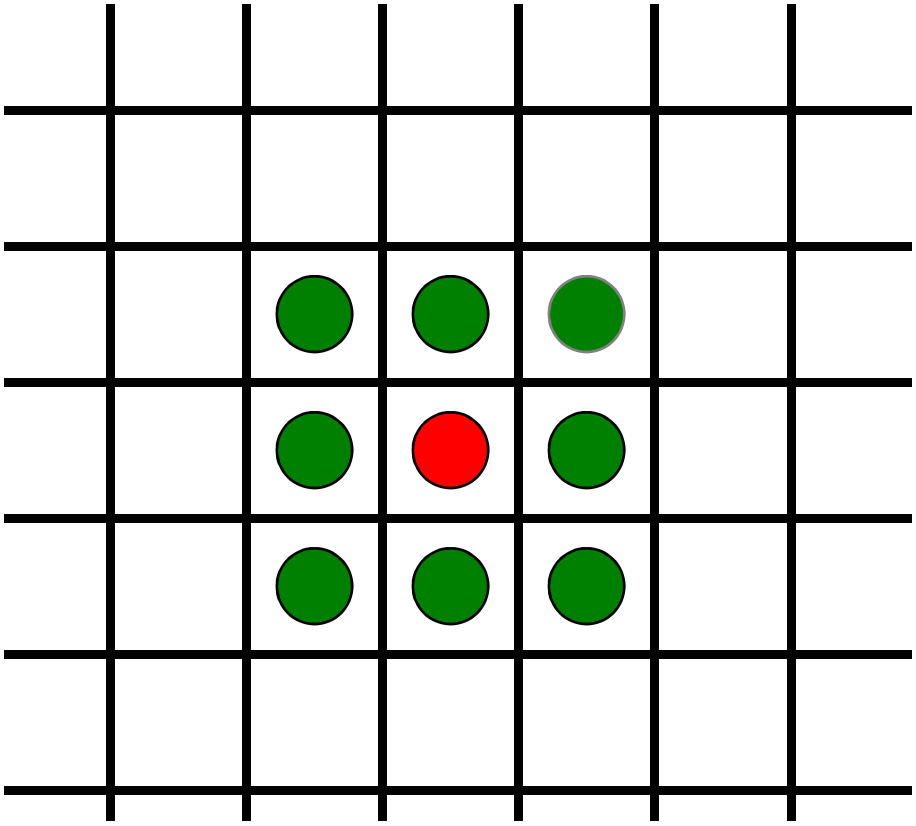
Шаг: 1

Клеточные автоматы

Пусть в ячейке **рождается** клетка, если в окрестности **1,2 или 3** другие клетки

Пусть в ячейке **выживает** клетка, если в окрестности **3,4 или 5** другие клетки

Это правило записывается **B123/S345**



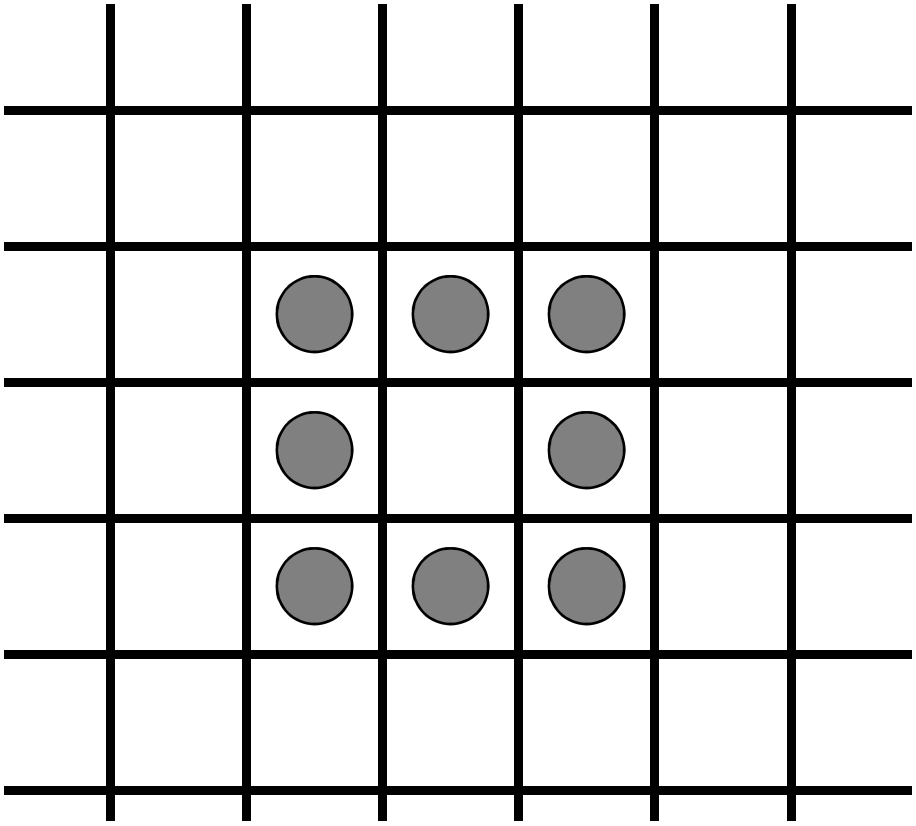
Шаг: 2 – рождение и выживание

Клеточные автоматы

Пусть в ячейке **рождается** клетка, если в окрестности **1,2 или 3** другие клетки

Пусть в ячейке **выживает** клетка, если в окрестности **3,4 или 5** другие клетки

Это правило записывается **B123/S345**



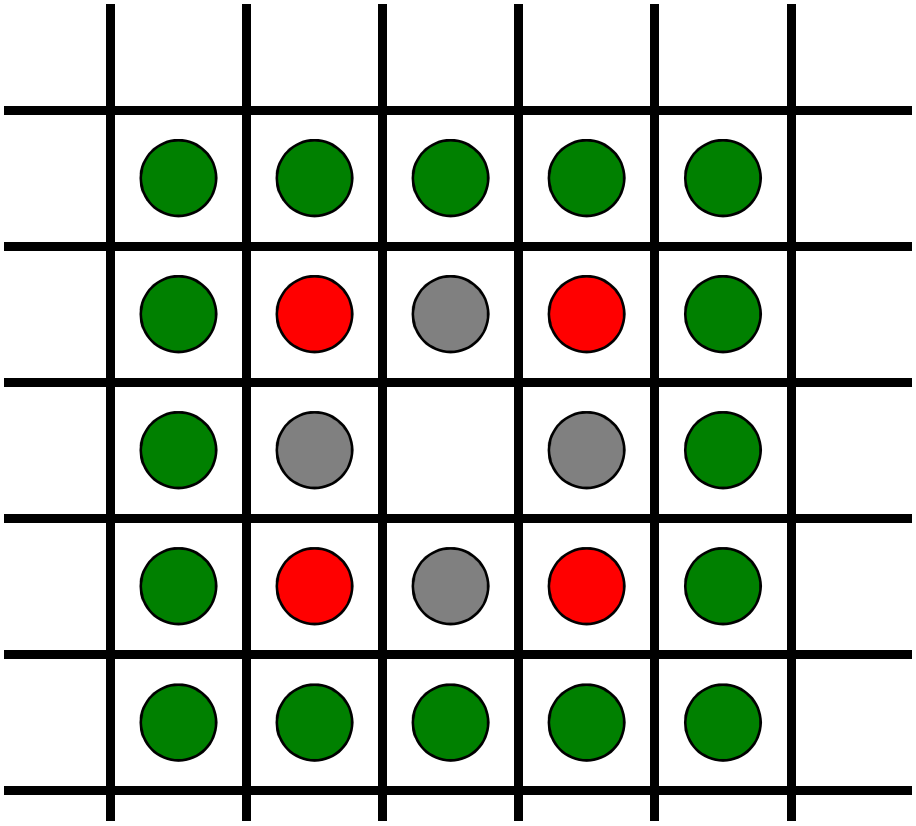
Шаг: 2 - итог

Клеточные автоматы

Пусть в ячейке **рождается** клетка, если в окрестности **1,2 или 3** другие клетки

Пусть в ячейке **выживает** клетка, если в окрестности **3,4 или 5** другие клетки

Это правило записывается **B123/S345**



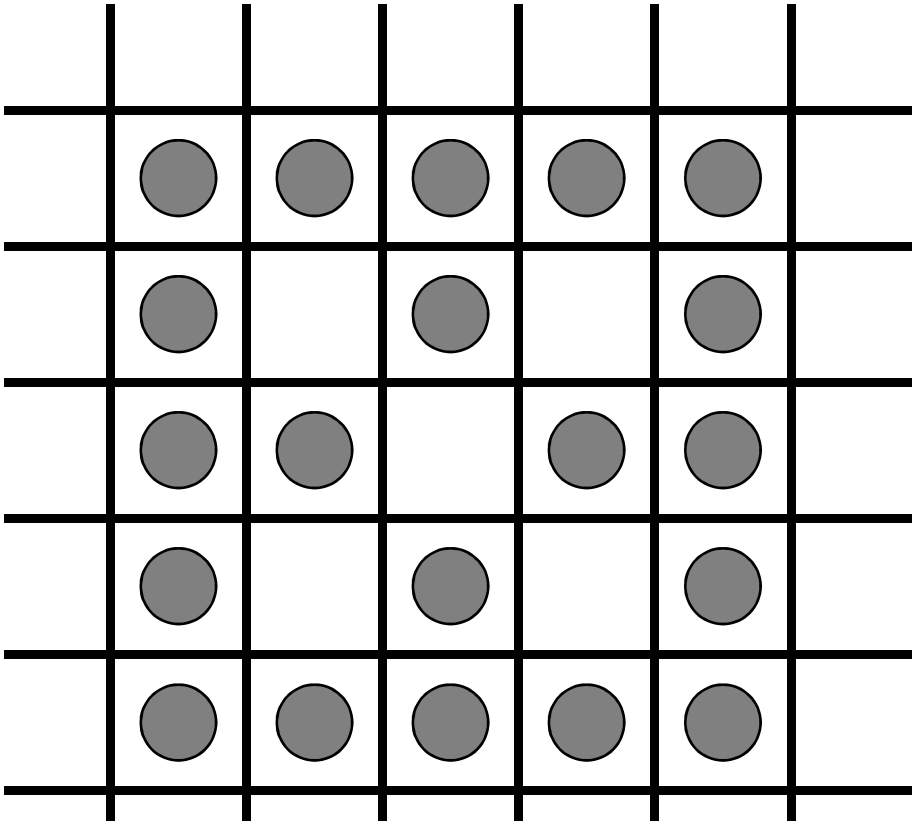
Шаг: 3 – рождение и выживание

Клеточные автоматы

Пусть в ячейке **рождается** клетка, если в окрестности **1,2 или 3** другие клетки

Пусть в ячейке **выживает** клетка, если в окрестности **3,4 или 5** другие клетки

Это правило записывается **B123/S345**



Шаг: 3 - итог