Для презентации

**2 слайд.** Существует довольно эффективный способ убедиться, что заданное число является составным, не разлагая это число на множители. Согласно малой теореме Ферма, если число N простое, то для любого целого a, не делящегося на N, выполняется сравнение:

Если же при каком-то a это сравнение нарушается, то можно утверждать, что N – составное. Вопрос только в том, как найти для составного N целое числа a, не удовлетворяющее данной формуле.

**3 слайд.** **Числа Кармайкла.** Для поиска чисел a мы можем испытывать все числа подряд на отрезке от 2 до N. Либо выбирать числа в этом промежутке случайным образом до тех пор, пока не найдем такое число, которое бы не удовлетворяло нашему условию (т.е. число N окажется составным), либо пока не переберем все возможные числа. Однако, такой подход может дать неправильный результат.

**4 слайд.** Существуют такие составные числа, которые будут удовлетворять условию. Такие числа называются числами Кармайкла. Для примера рассмотрим число 561 = 3\*11\*17.

**5 слайд.** Таким образом, получаем вид чисел Кармайкла: , где - простые, различные числа, , причем делится на каждую разность . Для устранения этого исключения, Генри Миллер в 1976 г. предложил несколько другую проверку.

**6 слайд.** А именно: если N – простое число, 𝑁−1=2^𝑠∗𝑡, где t нечетно, а s максимально возможное, то согласно малой теореме Ферма для каждого a с условием (a, N) = 1, хотя бы одна из скобок в произведении

(𝑎^𝑡−1)∗(𝑎^𝑡+1)∗(𝑎^2𝑡+1)∗...∗(𝑎^(2𝑠−1)+1)=𝑎^(𝑁−1)−1 делится на N.

Это свойство можно использовать для того, чтобы отличать составные числа от простых.

**7 слайд.** Для дальнейшего формулирования алгоритма, отличающего простые числа от составных, введем понятие:

пусть N – нечетное составное число, 𝑁−1=2^𝑠∗𝑡, где t нечетно. Назовем целое число a, 1 < 𝑎 <𝑁, «хорошим» для N, если нарушается одно из двух условий:

α) N не делится на a

β) 𝑎^𝑡≡1 (𝑚𝑜𝑑 𝑁) или существует целое k, 0≤ 𝑘 <s, такое, что

𝑎^(2^𝑘∗𝑡) ≡1 (𝑚𝑜𝑑 𝑁)

Из сказанного ранее следует, что для простого числа N не существует хороших чисел a. Если же N составное, то, как доказал Рабин, их существует не менее . Теперь, основываясь на условие Миллера, построим алгоритм, отличающий простые числа от составных.

**8 слайд.** 1. Выберем случайным образом число a, , и проверим для этого числа указанные выше свойства:

α) N не делится на a

β) или существует целое k, , что

2. Если хотя бы одно условие нарушается, то N составное

3. Если выполнены оба условия, то возвращаемся к шагу 1 и выбираем другое число.

**9 слайд. Алгоритм Миллера.** Миллер предложил детерминированный алгоритм определения составных чисел, однако справедливость его результата зависит от недоказанной в настоящее время расширенной гипотезы Римана. Согласно этому алгоритму, достаточно проверить предыдущие условие для всех чисел на отрезку от 2 до f(N). Если какое-то из условий не выполняется, то число N составное, иначе оно либо простое, либо является степенью простого числа, что можно легко проверить.

**10 слайд. Функция f(N).** В 1952 году Анкени доказал, что для каждого простого числа q существует квадратичный невычет a, удовлетворяющий неравенствам при некоторой достаточно большой константе c (квадратичным вычетом для числа p называется каждое целое число r, для которого существует целое число x такое, что число x^2 – r делится на p. Целые числа, не являющиеся квадратичными вычетами для p, называются его квадратичными невычетами), т.е. в нашем случае: . Несколько позднее было доказано, что в качестве константы можно взять число 70, а еще позднее это число уменьшили до 2.

**11 слайд.** Теперь можно точно сформулировать сам алгоритм Миллера:

1. Проверить, выполняется ли равенство при некоторых Если выполняется, то N – составное число, и алгоритм останавливается.

2. Выполнить следующие шаги для всех :

а) Проверить условие a|N

б) Проверить условие

в) Выяснить, верно ли, что при некотором k, (наиб. k, для которого верно: )

3. Если мы дошли до этого шага, то N – простое число.

12 **слайд. Пример алгоритма Миллера для числа 19.**

**13 слайд.** **Пример для числа 19 (скрин программы)**

**14 слайд. Пример программы для составного числа 3579 (скрин программы)**

**15 слайд.** Однако у этого алгоритма есть недостаток: при увеличении проверяемого числа значительно увеличивается время выполнения алгоритма. Поэтому был предложен более эффективный алгоритм – алгоритм Миллера-Рабина.

**16 слайд.** Он является модификацией алгоритма Миллера, с помощью которого можно определить, является ли заданное число составным. Однако, в сравнении с тестом Миллера, этот алгоритм не может строго доказать простоту числа. Тем не менее этот тест часто используют в криптографии для получения больших простых чисел.

**17 слайд. Псевдокод.** На слайде представлен псевдокод программа. Числа a находились также при помощи алгоритма Миллера, перебором всех чисел меньших f. Для шагов 3 и 4 были написаны функции из длинной арифметики.

**18 слайд. Тест Миллера-Рабина.** Алгоритм Миллера-Рабина параметризуется количеством итераций (числом r). Рекомендуется брать число r порядка величины :

Пусть . Представим число N – 1 в виде , где d – нечетно. Тогда если N – простое число, то для любого , выполняется одно условий:

1.

2.

Если эти условия выполняются для некоторых чисел a (выбирающихся произвольно), то число N называют вероятно простым (для уменьшения вероятность ошибки следует выполнить проверки для нескольких чисел a).

**19 слайд. Пример алгоритма Миллера-Рабина для числа 221. Тот же слайд: Выводы.** Итак, сформулируем выводы по работе: алгоритм Миллера – детерминированный тест на простоту числа (Детерминированный алгоритм — алгоритмический процесс, который выдаёт уникальный и предопределённый результат для заданных входных данных.). Он основывается на том, что число N либо является степенью некоторого числа, либо любое число a в промежутке от 2 до f(N) является свидетелем простоты по Миллера, причем значение функции f(N) намного меньше самого числа N. Данный алгоритм основывается на расширенной гипотезе Римана и поэтому является условно доказанным. Алгоритм Миллера гораздо медленнее теста Миллера-Рабина, который используется криптографии для проверки числа на простоту за малое время и при этом малую вероятность того, что число на самом деле является составным.