Отчёт по лабораторной работе №5

Аветисян Давид Артурович

9 ноября 2024

РУДН, Москва, Россия

Цель работы

Познакомиться с вероятностными алгоритмами проверки чисел на простоту.

Тест Ферма на языке Julia

- 2-3 строки: задание числа, которое нужн проверить на простоту, и количество проверок.
- 5 строка: задание функции.
- 6 строка: цикл проверки выполняется k раз.
- 7 строка: берётся случайно число a в диапазоне $[1, n-1]$.
- 8 строка: проводим проверку условия, при невыполнении сразу завершаем работу.
- 9-13 строки: выводим результат, закрываем функцию.

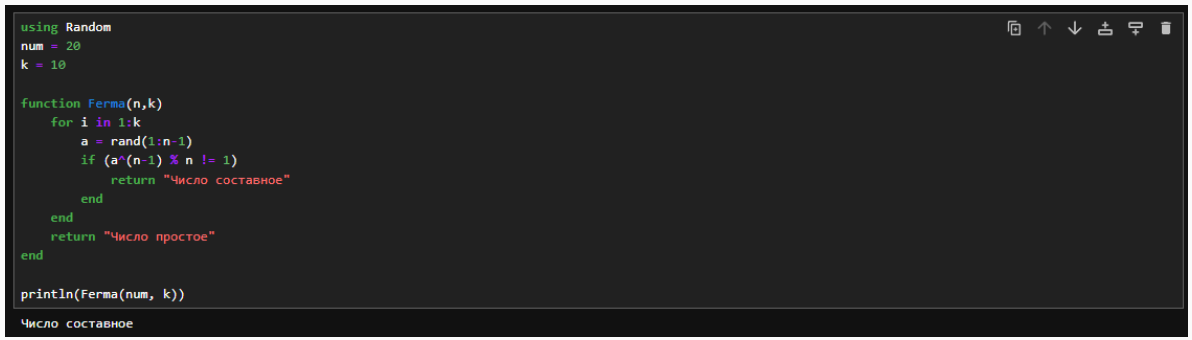


Рис. 1: Тест Ферма на языке Julia

Поиск символа Якоби на языке Julia

- 1-5 строки: задание функции, проверка условия для вычисления символа Якоби.
- 6-25 строки - реализация алгоритма: проверка трёх условия и действия согласно этим условиям: смена знака символа при четном и нечетном k, проверка остатков от деления.
- 26 строка: вывод результата раброты программы. В данном случае я вычислял Якоби 7 и 33. Вывод представлен на рисунке выше.

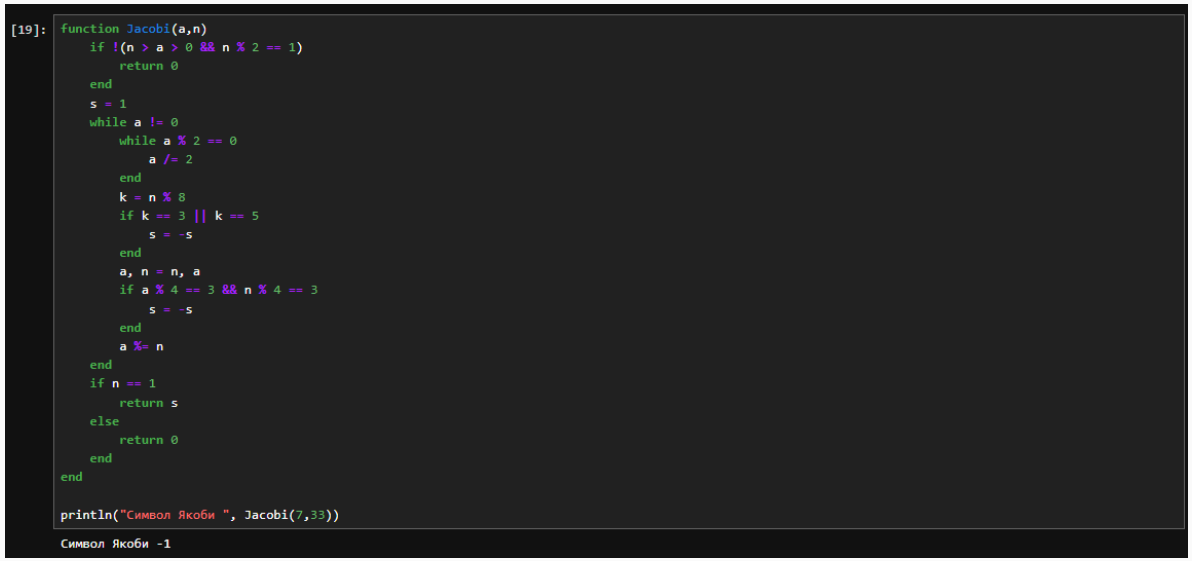
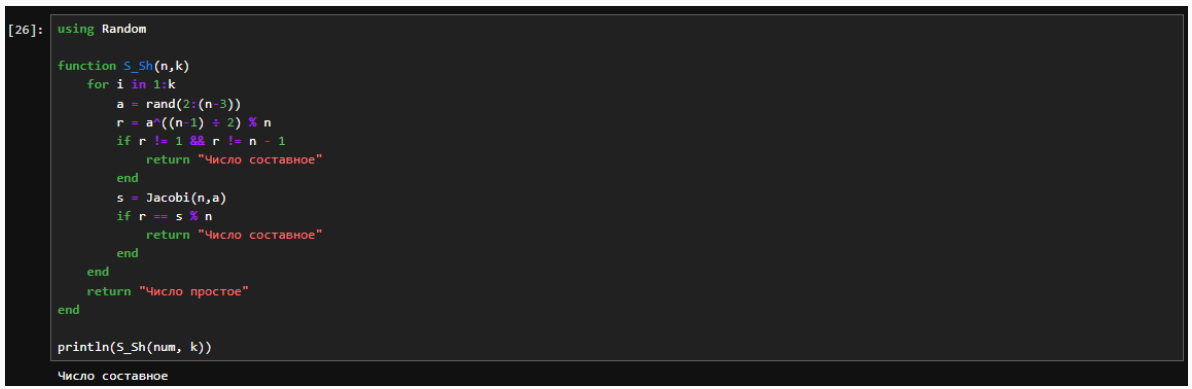


Рис. 2: Поиск символа Якоби на языке Julia

Тест Соловэя-Штрассена на языке Julia

- 3 строка: задаём функцию.
- 4 строка: повторим проверку k раз
- 5-16 строки - реализация алгоритма: выбираем случайное число a, вычисляем число r по формуле в строке 6, а затем проверяем получившееся значение на два условия. Если оно не проходит, проверку, то сразу заканчиваем работу программы. Далее следует ещё одна проверка условия в строке 11, при провале также завершаем работу.
- 18 строка: вывод на экран. Результат работы программы с числами 20 и 5 и 10-ю проверками.



Тест Миллера-Рабина на языке Julia

- 3 строка: задаём функцию.
- 4-9 строки: отсеивание числа 2 и остальных четных чисел.
- 11-29 строки - реализация алгоритма: выбираем случайное число a и вычисляем число x по формуле в строке 17. При условии в строке 18 выполняем дополнительные действия - вычисление остатка от деления квадра x на проверяемое число. Если число прошло все проверки k раз, мы определяем его как “вероятно, простое”.

```
function M_R(n, k)
    if n == 2
        return "Число простое"
    end
    if n % 2 == 0
        return "Число составное"
    end
    r, s = 0, n - 1
    while s % 2 == 0
        r += 1
        s += 2
    end
    for _ in 1:k
        a = rand(2:(n-1))
        x = powermod(a, s, n)
        if x == 1 || x == n - 1
            continue
        end
        for _ in 1:(r-1)
            x = powermod(x, 2, n)
            if x == n - 1
                break
            else
                return "Число составное"
            end
        end
    end
    return "Число простое"
end
```

Выводы

Я познакомился с вероятностными алгоритмами проверки чисел на простоту.