

Лабораторна работа № 6 - Вероятностные проверки чисел на простоту

Покрас Илья Михайлович НФИмд-01-24

29 сентября, 2024, Москва, Россия

Российский Университет Дружбы Народов Имени Патриса Лумумбы

Реализовать алгоритмы вероятностной проверки чисел на простоту.

- Реализовать алгоритм теста Ферма;
- Реализовать алгоритм вычисления символа Якоби;
- Реализовать алгоритм теста Соловья-Штрассена;
- Реализовать алгоритм теста Миллера-Рабина.

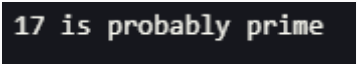
Алгоритм теста Ферма - код

```
function fermat_test(n)
    a = rand(2:n-2)
    if powermod(a,(n-1), n) != 1
        return "$n is composite"
    end
    return "$n is probably prime"
end

n = 17

fermat_test(n)
```

Рис. 1: Функция алгоритма теста Ферма



```
17 is probably prime
```

Рис. 2: Результат выполненного кода (1)

Алгоритм вычисления символа Якоби - код

```
function jacobi(a, n)
    if a == 0
        return 0
    elseif a == 1
        return 1
    end

    e = 0
    while a % 2 == 0
        a = a ÷ 2
        e += 1
    end

    if e % 2 == 0
        s = 1
    else
        s = n % 8 == 1 || n % 8 == 7 ? 1 : -1
    end

    if n % 4 == 3 && a % 4 == 3
        s = -s
    end

    a1 = a
    n1 = n % a1

    if a1 == 1
        return s
    else
        return s * jacobi(n1, a1)
    end
end

a = 2
n = 10
println(jacobi(a, n))
```

Рис. 3: Функция алгоритма вычисления символа Якоби

Алгоритм вычисления символа Якоби - результат выполнения



-1

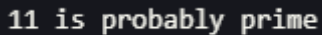
Рис. 4: Результат выполненного кода (2)

Алгоритм теста Соловья-Штрассена - код

```
> function jacobi(a, n) ...  
end  
  
function solovay_strassen(n)  
  
    a = rand(2:n-2)  
    r = a^((n-1)/2) % n  
    if r != 1 && r != n-1  
        return "$n is composite"  
    end  
    if r != jacobi(a, n) % n  
        return "$n is composite"  
    end  
  
    return "$n is probably prime"  
end  
  
n = 11  
println(solovay_strassen(n))
```

Рис. 5: Функция алгоритма теста Соловья-Штрассена

Алгоритм теста Соловья-Штрассена - результат выполнения



```
11 is probably prime
```

Рис. 6: Результат выполненного кода (3)

Алгоритм теста Миллера-Рабина - код

```
function miller_rabin(n)
    s = 0
    r = n - 1
    while r % 2 == 0
        r = r ÷ 2
        s += 1
    end

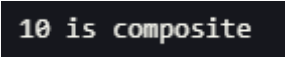
    a = rand(2:n-2)
    y = powermod(a, r, n)

    if y != 1 && y != n - 1
        j = 1
        while j <= s - 1 && y != n - 1
            y = (y^2) % n
            if y == 1
                return "$n is composite"
            end
            j += 1
        end
        if y != n - 1
            return "$n is composite"
        end
    end

    return "$n is probably prime"
end

n = 10
println(miller_rabin(n))
```

Рис. 7: Функция алгоритма теста Миллера-Рабина



```
10 is composite
```

Рис. 8: Результат выполненного кода (4)

Я реализовал алгоритмы вероятностной проверки чисел на простоту.

Спасибо за внимание
