

Вероятностные алгоритмы проверки чисел на простоту

Лабораторная работа №5

Данилова А.С.

Изучить вероятностные алгоритмы проверки чисел на простоту и реализовать их программно на языке Julia.

При проверке числа на простоту тестом Ферма выбирают несколько чисел a . Чем больше количество a , для которых утверждение истинно, тем больше вероятность, что число n простое.

Символ Якоби обобщает символ Лежандра на все нечётные числа, большие единицы. Символ Кронекера-Якоби, в свою очередь, обобщает символ Якоби на все целые числа, но в практических задачах символ Якоби играет гораздо более важную роль, чем символ Кронекера-Якоби.

Тест всегда корректно определяет, что простое число является простым, но для составных чисел с некоторой вероятностью он может дать неверный ответ. Основное преимущество теста заключается в том, что он, в отличие от теста Ферма, распознает числа Кармайкла как составные.

Тест Миллера-Рабина — вероятностный полиномиальный тест простоты. Тест Миллера-Рабина позволяет эффективно определять, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера-Рабина часто используется в криптографии для получения больших случайных простых чисел.

```
1  using Random
2
3  function ferm(n::Int)
4      a = rand(2:(n-2))
5      r = powermod(a, n - 1, n)
6
7      if r == 1
8          return "Число $n, вероятно, простое"
9      else
10         return "Число $n составное"
11     end
12 end
13
14 println("Введите число n")
15 n = parse{Int, readline()}
16
17 result = ferm(n)
18 println(result)
```

Рис. 1: Тест Ферма

Выполнение работы

```
1  using Random
2  function ja_sym(a::Int, n::Int)
3      g = 1
4      if a == 0
5          return 0
6      elseif a == 1
7          return g
8      end
9      a = a % n
10     while a != 0
11         k = 0
12         while a % 2 == 0
13             a = div(a, 2)
14             k += 1
15         end
16
17         if k % 2 == 0
18             s = 1
19         else
20             if n % 8 == 1 || n % 8 == 7
21                 s = 1
22             else
23                 s = -1
24             end
25         end
26         if a == 1
27             return g * s
28         end
29         if n % 4 == 3 && a % 4 == 3
30             s = -s
31         end
32         a = n % a
33         n = a
34         g *= s
35     end
36     return g
37 end
```



```
5/7 = -1
* Terminal will be reused by tasks, press any key to close it.

Activating new project at `C:\Users\nastd\.julia\environments\v1.11`
Введите a:
4
Введите n:
7
4/7 = 1
```

Рис. 3: Символ якоби от 4 и 7

```
function s_test(n::Int)
    a = rand(2:(n - 2))
    r = powermod(a, div(n - 1, 2), n)

    if r == 1 || r == n - 1
        return "Число $n, вероятно, простое"
    end

    return (r != ja_sym(a, n) % n) ? "Число $n составное" : "Число $n, вероятно, простое"
end

println("Введите n:")
n = parse{Int, readline()}
println(s_test(n))
```

Рис. 4: Тест Соловья-Штрассена

Выполнение работы

```
1 using Random
2 function miller_rabin(n::Int, k::Int)
3     d = n - 1
4     s = 0
5     while d % 2 == 0
6         d = Int(d / 2)
7         s += 1
8     end
9
10    for _ in 1:k
11        a = rand(2:(n - 2))
12        y = powermod(a, d, n)
13        if y == 1 || y == n - 1
14            continue
15        end
16
17        for _ in 1:s
18            y = powermod(y, 2, n)
19            if y == 1
20                return "Число $n составное"
21            elseif y == n - 1
22                break
23            end
24        end
25
26        if y != n - 1
27            return "Число $n составное"
28        end
29    end
30
31    return "Число $n простое"
32 end
33 println("Введите число:")
34 n = parse{Int, readline()}
35 k = 10
36 println(miller_rabin(n, k))
```

```
• Activating new project at `C:\Users\nastd\.julia\environments\v1.11`  
Введите n:  
31  
Число 31, вероятно, простое  
* Terminal will be reused by tasks, press any key to close it.  
  
• Activating new project at `C:\Users\nastd\.julia\environments\v1.11`  
Введите n:  
33  
Число 33 составное  
* Terminal will be reused by tasks, press any key to close it.
```

Рис. 6: Результаты

Мы изучили вероятностные алгоритмы проверки чисел на простоту и реализовали их программно на языке Julia.