Лабораторная работа №5

Вероятностные алгоритмы проверки чисел на простоту

Кубасов В.Ю.

Содержание

# 1 Цель работы

Реализовать предложенные вероятностные алгоритмы

# 2 Задание

Реализовать алгоритмы:  
- тест Ферма - символ якоби - тест Соловэя-Штрассена - тест Миллера-Рабина # Теоретическое введение

Ввиду возросшего интереса к простым числам и несуществования алгоритма с приемлемой[1 ] временной сложностью для разложения чисел на простые применяются вероятностные алгоритмы, которые в значительно меньшее число итераций позволяют оценить вероятность “простоты” числа. Самый используемый в современности - алгоритм Миллера-Рабина[2].

# 3 Выполнение лабораторной работы

## 3.1 Тест Ферма

println("Введите n");  
n = parse(Int, chomp(readline()));  
  
a = rand(2:n-2);  
  
r = a ^ (n - 1) % n;  
if (r == 1)  
 println("Число, вероятно, простое");  
else  
 println("Число составное");  
end;

## 3.2 Определения числа Якоби

function jackobi(n, a, g = 1)  
 if (a == 1)  
 return 0;  
 end;  
 if (a == 1)  
 return g;  
 end;  
 k = 0;  
 a1 = a;  
 while (a1 % 2 == 0)  
 a1 /= 2;  
 k += 1;  
 end;  
  
 s = 0;  
 if ((k % 2 == 0) || (abs(n % 8) == 1))  
 s = 1;  
 elseif (abs(n % 8) == 3)  
 s = -1  
 end;  
  
 if (a1 == 1)  
 return g \* s;  
 end;  
  
 if ((n % 4 == 3) && (a1 % 4 == 3))  
 s = -s;  
 end;  
  
 return jackobi(a1, n % a1, g \* s);   
end  
  
println(jackobi(91, 15));

## 3.3 Тест Соловэя-Штрассена

println("Введите n");  
n = parse(Int, chomp(readline()));  
  
a = rand(2:n-2);  
  
r = a ^ (n - 1) % 2;  
  
if ((r != 1) && (r != n - 1))  
 println("Число n составное");  
else  
  
 s = jackobi(n, a);  
   
 if (r % n == s)  
 println("Число n составное");  
 else   
 println("Число, вероятно, простое");  
 end;  
  
end;

## 3.4 Тест Миллера-Рабина

function miller()  
 println("Введите n");  
 n = parse(Int, chomp(readline()));  
   
 n\_1 = n - 1;  
 s = 0;  
   
 while (n\_1 % 2 == 0)  
 n\_1 /= 2;  
 s += 1;  
 end;  
   
 r = n\_1;  
   
 a = rand(2:n-2);  
 y = a^r % n;  
   
 j = 1;  
 while (y != 1 && y != n - 1)  
 if ((j <= s - 1) && (y != n - 1))  
 y = y ^ 2 % n;  
 if (y == 1)  
 println("Число n составное");  
 return 0;  
 break;  
 end;  
 j += 1;  
 end;  
   
 if (y != n - 1)  
 println("Число n составное");  
 return 0;  
 break;  
 end;  
 end;  
   
 println("Число n, вероятно, простое")  
end;  
  
miller();

# 4 Выводы:

* В ходе лабораторной работы реализовали вероятностные алгоритмы определения числа на простоту

# Список литературы

1. Коломийцева С., Соколова К. Сравнительный анализ алгоритмов проверки чисел на простоту // Информационные технологии и высокопроизводительные вычисления. 2019. С. 90–96.

2. Бердимуратов М.К., Ибрагимов К. АЛГОРИТМ МИЛЛЕРА-РАБИНА ДЛЯ ПРОВЕРКИ ЧИСЕЛ НА ПРОСТОТУ // ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ. 2023. Т. 35, № 1. С. 51–53.