Отчёт по лабораторной работе №5

Вероятностные алгоритмы проверки чисел на простоту

Шутенко Виктория Михайловна

Содержание

# 1 Цель рабoты

Приoбрести практические навыки рабoты с вероятностными алгоритмами проверки чисел на простоту.

# 2 Хoд рабoты

## 2.1 Реализация теста Ферма

def fermats(n: int):  
 if n < 5:  
 print("N should be greater or equal to 5")  
 return  
   
 a = np.random.choice(range(2, n-1))  
   
 if (a\*\*(n-1)) % n == 1:  
 return "Число " + str(n) + ", вероятно, простое"  
 else:  
 return "Число " + str(n) + " составное"

## 2.2 Реалиазация символа Якоби

def yakobi(n: int, a: int):  
 if n < 3:  
 print("Число n должно быть больше или равно 3")  
 return None  
   
 if a < 0 or a >= n:  
 print("Число a должны быть на интервале [0;n)")  
 return None  
   
 g = 1  
   
 while a != 0 and a != 1:  
 k = 0  
 a\_1 = a  
  
 while divmod(a\_1, 2)[1] != 1:  
 a\_1 = divmod(a\_1, 2)[0]  
  
 while (2\*\*k)\*a\_1 != a:  
 k += 1  
   
 s = 1  
 if k % 2 == 0:  
 s = 1  
 else:  
 if (n == 1 % 8) or (n == -1 % 8):  
 s = 1  
 elif (n == 3 % 8) or (n == -3 % 8):  
 s = -1  
   
 if a\_1 == 1:  
 return g \* s  
   
 if (n == 3 % 4) and (a\_1 == 3 % 4):  
 s = -s  
  
 a = n % a\_1  
 n = a\_1  
 g = g \* s  
  
 if a == 0:  
 return 0  
 else:  
 return g

## 2.3 Реализация теста Соловэя-Штрассена

def soloway\_shtrassen(n: int):  
 if n % 2 == 0 or n < 5:  
 return "Число " + str(n) + " составное"  
   
 a = np.random.randint(2, n-1)  
 r = int((a\*\*((n-1)/2)) % n)  
   
 if r != 1 and r != (n - 1):  
 return "Число " + str(n) + " составное"  
   
 s = yakobi(n, a)  
 if r == s % n:  
 return "Число " + str(n) + " составное"  
 else:  
 return "Число " + str(n) + ", вероятно, простое"

## 2.4 Реализация теста Миллера-Рабина

def miller\_rabin(n: int):  
 if n % 2 == 0 or n < 5:  
 return "Число " + str(n) + " составное"  
   
 r = n - 1  
 s = 0  
   
 while divmod(r, 2)[1] != 1:  
 r = divmod(r, 2)[0]  
  
 while (2\*\*s)\*r != n-1:  
 s += 1  
   
 a = np.random.randint(2, n-1)  
 y = (a\*\*r) % n  
   
 if y != 1 and y != n - 1:  
 j = 1  
 while j <= s - 1 and y != n - 1:  
 y = (y\*\*2) % n  
 if y == 1:  
 return "Число " + str(n) + " составное"  
 j = j + 1  
 if y != n - 1:  
 return "Число " + str(n) + " составное"  
   
 return "Число " + str(n) + ", вероятно, простое"

## 2.5 Тестирование

def main():  
 n = [7, 11, 13, 17, 19, 23, 27, 29, 31, 33, 35]  
   
 for n\_i in n:  
 print(f'\n----ЧИСЛО-{n\_i}----\n')  
 print(f'Тест Ферма: {fermats(n\_i)}')  
 print(f'Тест Соловэя-Штрассена: {soloway\_shtrassen(n\_i)}')  
 print(f'Тест Миллера-Рабина: {miller\_rabin(n\_i)}')

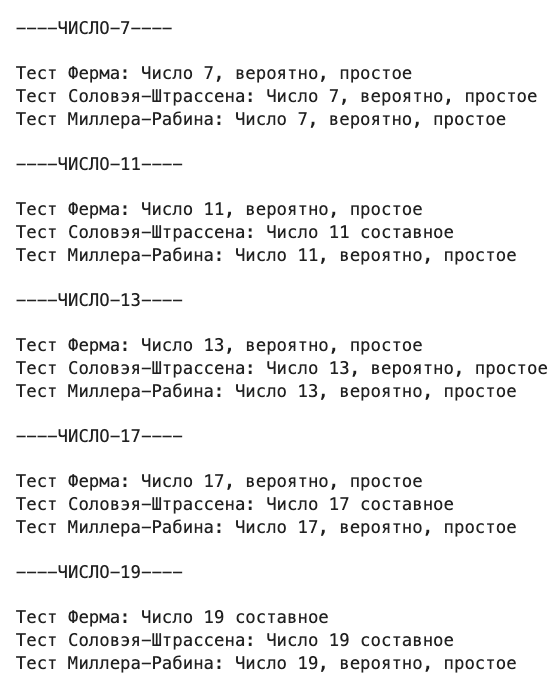


Figure 1: Тестирование

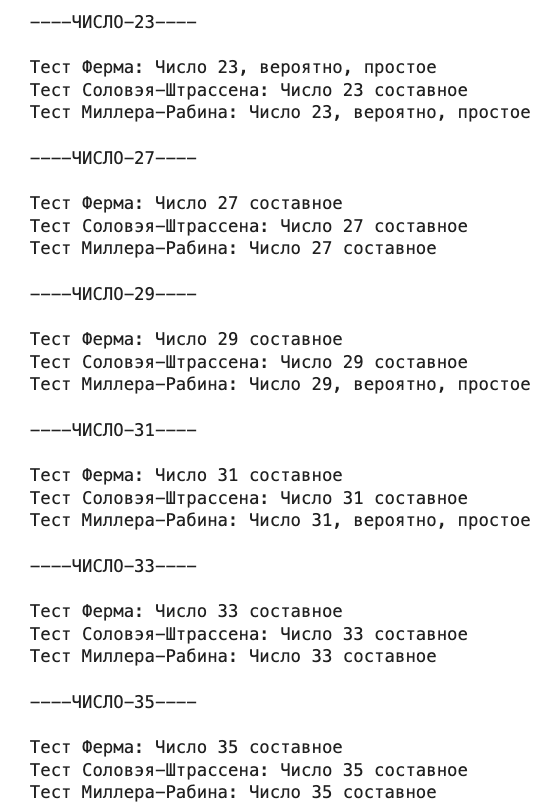


Figure 2: Тестирование