



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 10
по курсу «Численные методы линейной алгебры»
«Реализация метода А.А.Карацубы»

Студент группы ИУ9-71Б Окутин Д.А.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

1. Реализовать метод А.А.Карацубы для умножения чисел.
2. Сравнить скорость работы с обычным умножением чисел.

2 Реализация

Исходный код программы представлен в листинге 1.

Листинг 1: code

```
1 using BenchmarkTools
2
3 function karatsuba(x::Int, y::Int)::Int
4     if x < (1 << 32) || y < (1 << 32)
5         return naive_multiplication(x,y)
6     end
7
8     n = max(floor(Int, log2(x) + 1), floor(Int, log2(y) + 1))
9     half = div(n, 2)
10
11     high_x, low_x = x >> half, x & ((1 << half) - 1)
12     high_y, low_y = y >> half, y & ((1 << half) - 1)
13
14     z0 = karatsuba(low_x, low_y)
15     z1 = karatsuba((low_x + high_x), (low_y + high_y))
16     z2 = karatsuba(high_x, high_y)
17
18     return (z2 << (2 * half)) + ((z1 - z2 - z0) << half) + z0
19 end
20
21 function naive_multiplication(x::Int, y::Int)::Int
22     x_digits = digits(x, base=2)
23     y_digits = digits(y, base=2)
24
25     result = 0
26     for i in 1:length(y_digits)
27         for j in 1:length(x_digits)
28             result += (x_digits[j] * y_digits[i]) << (i + j - 2)
29         end
30     end
31
32     return result
33 end
34
```

```

35
36 function generate_large_number(bits::Int)::Int
37     return rand(2^(bits-1):2^bits - 1)
38 end
39
40 using PyPlot
41 regular_times = []
42 karatsuba_times = []
43 D = [1, 2, 4, 8, 16, 32, 36, 40, 44, 48]
44
45
46 for digits in D
47     x = generate_large_number(digits)
48     y = generate_large_number(digits)
49
50     regular_time = @belapsed naive_multiplication($x, $y)
51     karatsuba_time = @belapsed karatsuba($x,$y)
52
53     # println("                : $digits")
54     # println("                : $regular_time
55     .*")
56     # println("                : $karatsuba_time
57     .*")
58
59     push!(regular_times, regular_time)
60     push!(karatsuba_times, karatsuba_time)
61
62     # println(karatsuba(x,y) == x * y)
63 end
64
65 PyPlot.figure(figsize=(6, 5))
66 PyPlot.title("
67
68 ")
69 PyPlot.xlabel("n")
70 PyPlot.ylabel("time")
71 PyPlot.plot(D, regular_times, label="
72
73 ")
74 PyPlot.plot(D, karatsuba_times, label="
75
76 ")
77 PyPlot.grid()
78 PyPlot.legend()
79 PyPlot.show()

```

3 Результаты

Результаты запуска представлены на рисунках 1.

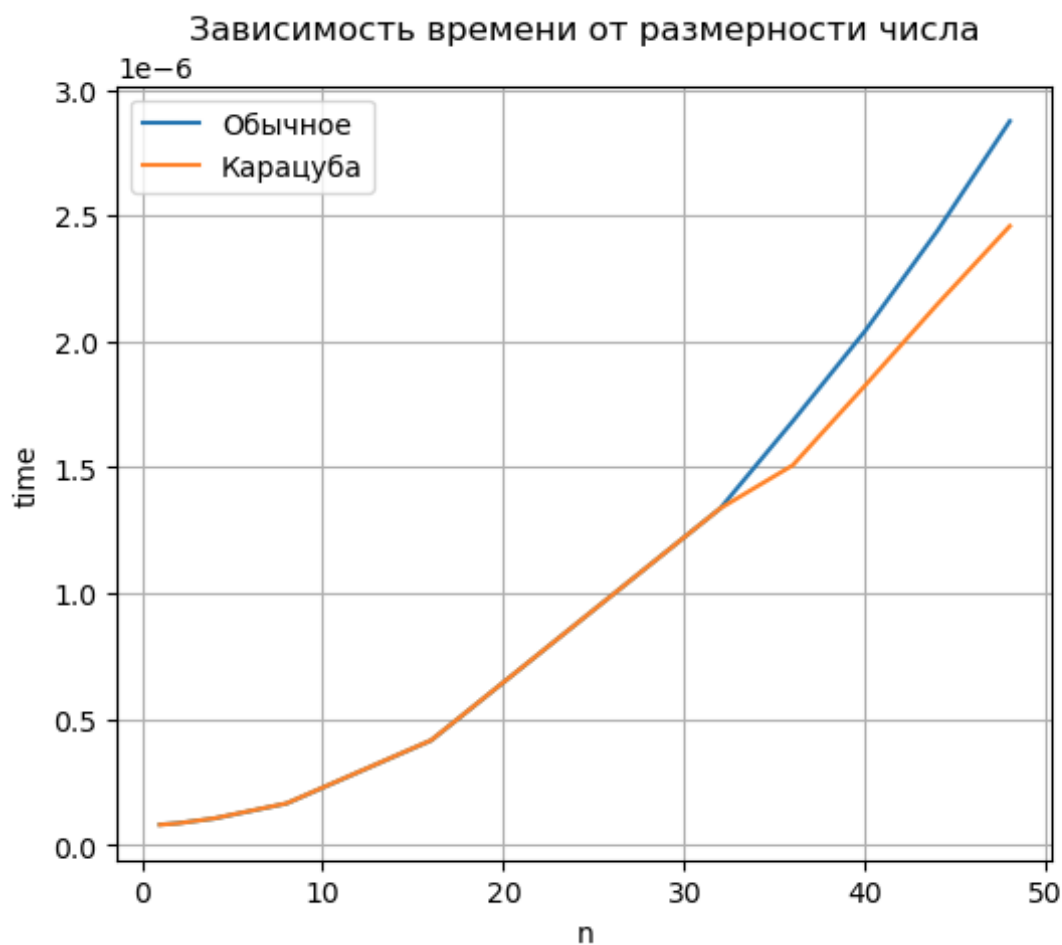


Рис. 1 — Результаты

4 Выводы

В результате выполнения данной лабораторной работы был реализован метод А.А.Карацубы для умножения чисел. Были проведены замеры времени для демонстрации эффективности алгоритма.