Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования   
«Алтайский государственный техничеcкий университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной работе № 8

"РЕКУРСИВНЫЕ ВЫЧИСЛЕНИЯ. МЕМОИЗАЦИЯ"

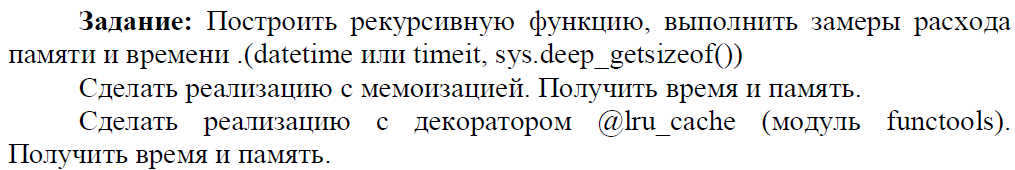
по дисциплине «Функциональное и логическое программирование»

Студент группы ПИ-91 Шинтяпин И. И.

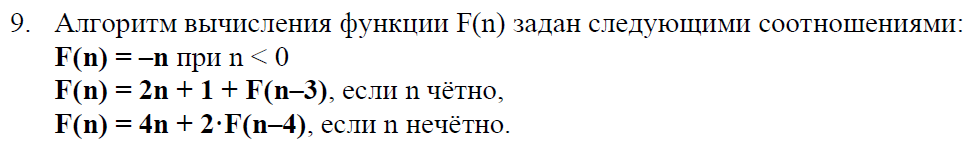
Преподаватель доцент, к.ф.-м.н. Андреева А.Ю.

Барнаул 2022

Номер в списке - 28



**Вариант 9:**



**Решение:**

from functools import lru\_cache

import tracemalloc

from time import time

# Обычная рекурсия

def F1(n):

if n < 0:

return -n

if n % 2 == 0:

return 2 \* n + 1 + F1(n - 3)

else:

return 4 \* n + 2 \* F1(n - 4)

# Мемоизация

dict\_storage = {}

def F2(n):

if dict\_storage.get(n) is not None:

return dict\_storage.get(n)

elif n < 0:

dict\_storage[n] = -n

elif n % 2 == 0:

dict\_storage[n] = 2 \* n + 1 + F2(n - 3)

else:

dict\_storage[n] = 4 \* n + 2 \* F2(n - 4)

return dict\_storage[n]

# Декоратор

@lru\_cache

def F3(n):

if n < 0:

return -n

if n % 2 == 0:

return 2 \* n + 1 + F3(n - 3)

else:

return 4 \* n + 2 \* F3(n - 4)

# Main

#Простая рекурсия

print("n = 1000")

tracemalloc.start()

startTime = time()

F1(1000)

endTime = time()

current, peak = tracemalloc.get\_traced\_memory()

print(f"\n\033[37mМетод решения:\033[35;1m {'Рекурсивная функция'}\033[0m")

print(f"\033[37mВремя:\033[36m {endTime - startTime} C\033[0m")

print(f"\033[37mPeak :\033[36m {peak / (10 \*\* 3)}КБ\033[0m")

tracemalloc.stop()

#Мемоизация

tracemalloc.start()

startTime = time()

F2(1000)

endTime = time()

current, peak = tracemalloc.get\_traced\_memory()

print(f"\n\033[37mМетод решения:\033[35;1m {'Мемоизация'}\033[0m")

print(f"\033[37mВремя:\033[36m {endTime - startTime} С\033[0m")

print(f"\033[37mPeak :\033[36m {peak / (10 \*\* 3)}КБ\033[0m")

tracemalloc.stop()

#@lru\_cache

tracemalloc.start()

startTime = time()

F3(1000)

endTime = time()

current, peak = tracemalloc.get\_traced\_memory()

print(f"\n\033[37mFunction Name :\033[35;1m {'@lru\_cache'}\033[0m")

print(f"\033[37mВремя:\033[36m {endTime - startTime} C\033[0m")

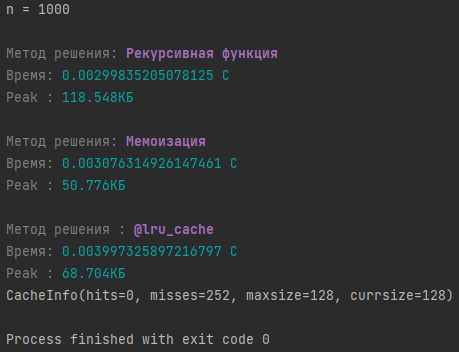
print(f"\033[37mPeak :\033[36m {peak / (10 \*\* 3)}КБ\033[0m")

print(F3.cache\_info())

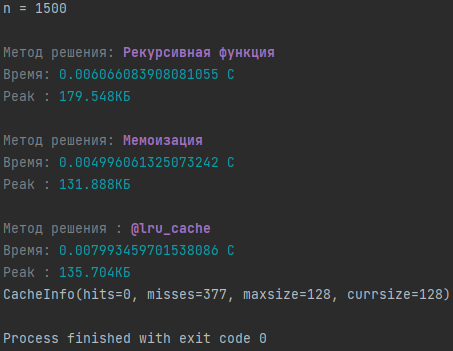
tracemalloc.stop()

**Тесты:**

1. N=1000



1. N=1500



1. N=500

