

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Языки программирования
Отчет по лабораторной работе №2**

Работа с
множествами в языке Python

Выполнил студент группы

ИТС-б-о-20-1

Скрыпник А.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент
Кафедры инфокоммуникаций, старший
преподаватель
Воронкин Р.А.

(подпись)

Ставрополь 2021

Цель работы: приобретение навыков по работе с рекурсивными функциями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Ссылка на репозиторий:

1) Создал приватный репозиторий LR4-2 - <https://github.com/Skr1P1/LR4-2.git>.

2) Решил задание: Самостоятельно изучите работу со стандартным пакетом Python timeit. Оцените с помощью этого модуля скорость работы итеративной и рекурсивной версий функций factorial и fib . Во сколько раз измениться скорость работы рекурсивных версий функций factorial и fib при использовании декоратора lru_cache.

```
PS F:\S_P\Учёба\2 курс\ЯП\ЛР4> f.; cd 'f:\S_P\Учёба\2 курс\ЯП\ЛР4'; & 'C:\Users\SerJ\AppData
sions\ms-python.python-2021.11.1422169775\pythonFiles\lib\python\debugpy\launcher' '49618'
Факториал без оптимизации выполняется за 0.00062989999999985 секунд
Факториал с оптимизацией выполняется за 0.000461299999999978 секунд
6765
Фиббоначи без оптимизации выполняется за 0.010066300000000084 секунд
6765
Фиббоначи с оптимизацией выполняется за 0.00507169999999979 секунд
PS F:\S_P\Учёба\2 курс\ЯП\ЛР4> █
```

Рисунок 1. Пример из методического указания

3) Решил индивидуальное задание 19(5): Создайте рекурсивную функцию, печатающую все подмножества множества $\{1, 2, \dots, N\}$.

```
PS F:\S_P\Учёба\2 курс\ЯП\ЛР4> & 'C:\Users\SerJ\AppData\Local\Mi
9775\pythonFiles\lib\python\debugpy\launcher' '62370' '--' 'f:\S_
[[], [3], [2], [2, 3], [1], [1, 3], [1, 2], [1, 2, 3]]
PS F:\S_P\Учёба\2 курс\ЯП\ЛР4>
```

Рисунок 2. Индивидуальная задача №5 из методического указания

Вывод: в ходе лабораторной работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.

Вопросы для защиты работы:

1. Для чего нужна рекурсия?

Рекурсия занимает меньше времени и может работать в обратную сторону.

2. Что называется базой рекурсии?

Выражение, которое её останавливает.

3. Как получить текущее значение максимальной глубины рекурсии в языке Python?

Командой «`sys.getrecursionlimit()`».

4. Что произойдет если число рекурсивных вызовов превысит максимальную глубину рекурсии в языке Python?

Будет ошибка «`RecursionError`».

5. Как изменить максимальную глубину рекурсии в языке Python?

Командой «`sys.setrecursionlimit(limit)`».

6. Каково назначение декоратора `lru_cache`?

Его используют для уменьшения количества лишних вычислений.

7. Что такое хвостовая рекурсия?

Хвостовая рекурсия — частный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции.