

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №7
дисциплины «Алгоритмизация»

Выполнил:
Гайчук Дарья Дмитриевна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.-доцент кафедры
инфокоммуникаций

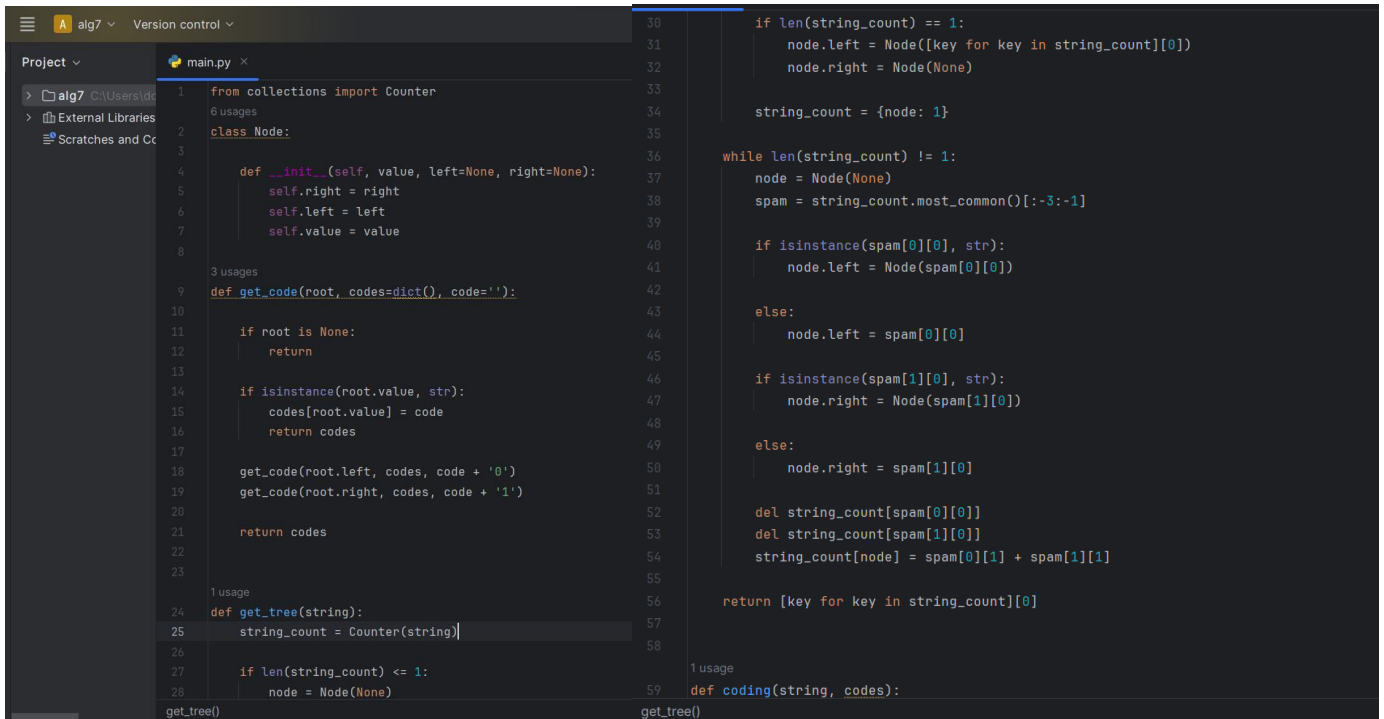
(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

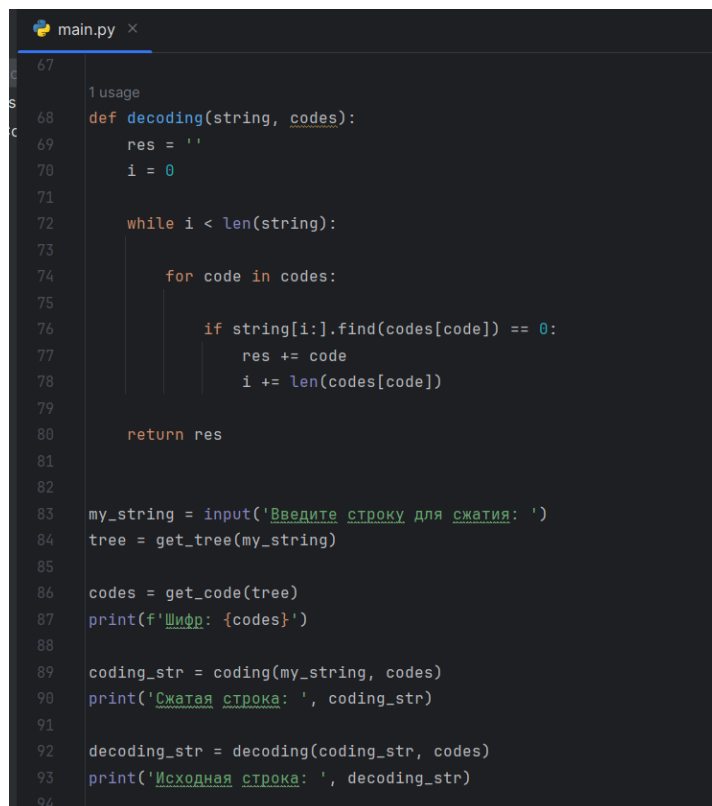
Порядок выполнения работы:

1. Написала программу кодирования и декодирования текста при помощи кодировки Хаффмана:



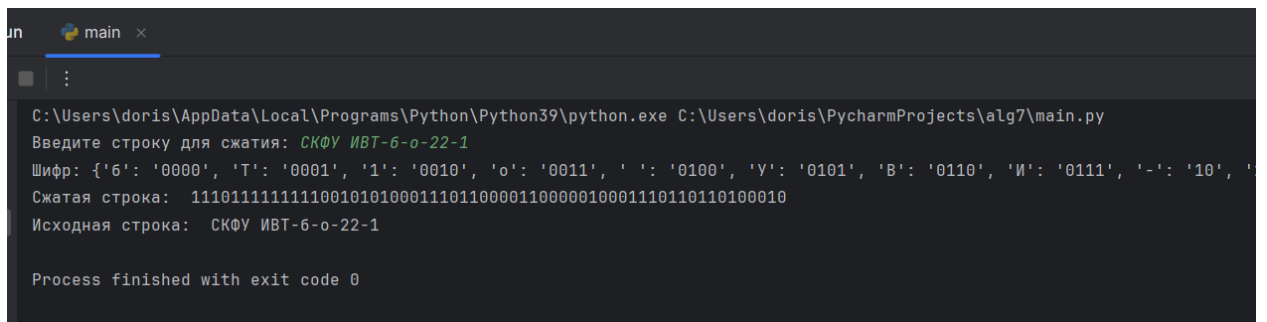
```
1 from collections import Counter
2
3 class Node:
4     def __init__(self, value, left=None, right=None):
5         self.right = right
6         self.left = left
7         self.value = value
8
9     3 usages
10    def get_code(root, codes=dict(), code=''):
11        if root is None:
12            return
13
14        if isinstance(root.value, str):
15            codes[root.value] = code
16            return codes
17
18        get_code(root.left, codes, code + '0')
19        get_code(root.right, codes, code + '1')
20
21        return codes
22
23    1 usage
24    def get_tree(string):
25        string_count = Counter(string)
26
27        if len(string_count) <= 1:
28            node = Node(None)
29
30            if len(string_count) == 1:
31                node.left = Node([key for key in string_count][0])
32                node.right = Node(None)
33
34            string_count = {node: 1}
35
36        while len(string_count) != 1:
37            node = Node(None)
38            spam = string_count.most_common()[::-3:-1]
39
40            if isinstance(spam[0][0], str):
41                node.left = Node(spam[0][0])
42
43            else:
44                node.left = spam[0][0]
45
46            if isinstance(spam[1][0], str):
47                node.right = Node(spam[1][0])
48
49            else:
50                node.right = spam[1][0]
51
52            del string_count[spam[0][0]]
53            del string_count[spam[1][0]]
54            string_count[node] = spam[0][1] + spam[1][1]
55
56        return [key for key in string_count][0]
57
58    1 usage
59    def coding(string, codes):
60
61    def decoding(string, codes):
62        res = ''
63        i = 0
64
65        while i < len(string):
66            for code in codes:
67                if string[i:].find(code) == 0:
68                    res += code
69                    i += len(code)
70
71        return res
72
73    my_string = input('Введите строку для сжатия: ')
74    tree = get_tree(my_string)
75
76    codes = get_code(tree)
77    print(f'Шифр: {codes}')
78
79    coding_str = coding(my_string, codes)
80    print('Сжатая строка: ', coding_str)
81
82    decoding_str = decoding(coding_str, codes)
83    print('Исходная строка: ', decoding_str)
```

Рисунок 1. Код программы



```
67
68 1 usage
69 def decoding(string, codes):
70     res = ''
71     i = 0
72
73     while i < len(string):
74         for code in codes:
75             if string[i:].find(code) == 0:
76                 res += code
77                 i += len(code)
78
79     return res
80
81
82
83 my_string = input('Введите строку для сжатия: ')
84 tree = get_tree(my_string)
85
86 codes = get_code(tree)
87 print(f'Шифр: {codes}')
88
89 coding_str = coding(my_string, codes)
90 print('Сжатая строка: ', coding_str)
91
92 decoding_str = decoding(coding_str, codes)
93 print('Исходная строка: ', decoding_str)
94
```

Рисунок 2. Вывод результата



```
main x
C:\Users\doris\AppData\Local\Programs\Python\Python39\python.exe C:\Users\doris\PycharmProjects\alg7\main.py
Введите строку для сжатия: СКФУ ИБТ-6-о-22-1
Шифр: {'6': '0000', 'T': '0001', '1': '0010', 'o': '0011', ' ': '0100', 'Y': '0101', 'B': '0110', 'И': '0111', '-': '10', '
Сжатая строка: 11101111111100101010001110110000110000010001110110110100010
Исходная строка: СКФУ ИБТ-6-о-22-1

Process finished with exit code 0
```

Рисунок 3. Результат работы программы

В ходе выполнения лабораторной работы был исследован метод кодирования Хаффмана: как с помощью него кодировать и декодировать текст. На основании этого исследования можно сделать следующий вывод: метод кодирования Хаффмана представляет собой эффективный способ сжатия данных, особенно текстовых, где некоторые символы встречаются чаще, чем другие.