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 ПРАКТИЧЕСКОЙ РАБОТЕ №7
дисциплины «Алгоритмизация»

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

_____
(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук, доцент, доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: алгоритм Хаффмана

Порядок выполнения работы:

Алгоритм:

1. Расчёт частоты символов

```
# Расчёт частоты символов
for char in sentence:
    if char in symbs:
        symbs[char] += 1
    else:
        symbs[char] = 1
for char, count in symbs.items():
    print(f"'{char}': {count} раз")
```

Рисунок 1 – Фрагмент кода, в котором считается частота символа

2. Построение дерева в соответствии с процедурой хаффмана

```
# Процедура построения дерева
def procedurehuffman(f):
    h = []
    buffer_fs = set()
    for i in f:
        heappush(h, (f[i], i))
    while len(h) > 1:
        f1, i = heappop(h)
        f2, j = heappop(h)
        fs = f1 + f2
        ord_val = ord('a')
        fl = str(fs)
        while fl in buffer_fs:
            letter = chr(ord_val)
            fl = str(fs) + " " + letter
            ord_val += 1
        buffer_fs.add(fl)
        f[fl] = {f"{x}": f[x] for x in [i, j]}
        del f[i], f[j]
        heappush(h, (fs, fl))
    return f
```

Рисунок 2 – Функция построения дерева «procedurehuffman»

3. Кодирование

```python
# Кодирование
def codingHuffman(sentence, dictionary):
    replaced_sentence = ''
    for char in sentence:
        if char in dictionary:
            replaced_sentence += dictionary[char]
        else:
            replaced_sentence += char
    return replaced_sentence
```

Рисунок 3 – Функция кодирования символов «codingHuffman»

4. Декодирование

```python
# Декодирование
def decodeHuffman(encoded_text, huffman_tree):
    decoded_text = ""
    key = list(huffman_tree.keys())[0]
    cur = huffman_tree[key]
    for bit in encoded_text:
        for i, (node, child) in enumerate(cur.items()):
            if str(i) != bit:
                if isinstance(child, int):
                    decoded_text += node
                    cur = huffman_tree[key]
                    break
                cur = child
                break
    return decoded_text
```

Рисунок 4 – Функция декодирования символов «codingHuffman»

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Лео\Desktop\gitalg\lab7-alg\prog> python3 .\haffman.py
Предложение: asdasdasdadas dsxgfdgfdsrd sdfsdfsdfsdfsfdsd sdfsdfjiosdj  jsdjfhsdfjsdfsd
'a': 5 раз
's': 19 раз
'd': 21 раз
' ': 5 раз
'x': 1 раз
'g': 2 раз
'f': 12 раз
'r': 1 раз
'j': 5 раз
'i': 1 раз
'o': 1 раз
'h': 1 раз
Дерево: {'74': {'31': {'f': 12, 's': 19}, '43': {'d': 21, '22': {'10': {' ': 5, 'a': 5}, '12': {'j': 5, '7': {'3': {'x': 1, '2': {'h': 1, 'i': 1}}, '4': {'2 a': {'o': 1, 'r': 1}, 'g': 2}}}}}}}
Коды: {'f': '11', 's': '10', 'd': '01', ' ': '0011', 'a': '0010', 'j': '0001', 'x': '000011', 'h': '0000101', 'i': '0000100', 'o': '0000011', 'r': '0000010', 'g': '000000'}
Закодированное предложение:0010100100101001001010010010010010100011011000001100000011010000001101100000010010011100111100111100111100111101101100100111001111001110001000010000000011100100010011001100011001000111000010
110011100011001111001
Раскодированное предложение:asdasdasdadas dsxgfdgfdsrd sdfsdfsdfsdfsfdsd sdfsdfjiosdj  jsdjfhsdfjsdfsd
PS C:\Users\Лео\Desktop\gitalg\lab7-alg\prog> python3 .\haffman.py
Предложение: Hello world!!!!!!
'H': 1 раз
'e': 1 раз
'l': 3 раз
'o': 2 раз
' ': 1 раз
'w': 1 раз
'r': 1 раз
'd': 1 раз
'!': 6 раз
Дерево: {'17': {'7': {'l': 3, '4': {'2': {' ': 1, 'H': 1}, '2 a': {'d': 1, 'e': 1}}}, '10': {'4 a': {'2 b': {'r': 1, 'w': 1}, 'o': 2}, '!': 6}}}
Коды: {'l': '11', ' ': '1011', 'H': '1010', 'd': '1001', 'e': '1000', 'r': '0111', 'w': '0110', 'o': '010', '!': '00'}
Закодированное предложение:101010001111010101101100100111111001000000000000
Раскодированное предложение:Hello world!!!!!!
PS C:\Users\Лео\Desktop\gitalg\lab7-alg\prog>
```

Рисунок 5 – Результат выполнения программы haffman.py

Вывод: в результате проделанной работы было выяснено как кодировать предложения с помощью алгоритма хаффмана.