

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 №6**  
**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:

Задача: написать алгоритм «Код Хаффмана», который на вход принимает частоты символов  $f_1, \dots, f_n \in N$ . А выводит строго двоичное дерево (у каждой вершины либо ноль, либо два сына), листья которого помечены частотами  $f_1, \dots, f_n$  минимизирующее  $\sum_{i=1}^n f_i \cdot (\text{глубина листа } f_i)$ .

### Алгоритм решения:

1. Задаётся очередь с приоритетами.
2. Добавляются частоты в очередь с приоритетами.
3. Перебирается  $k$  от длины массива частот  $+ 1$  до удвоенной длины массива частот  $- 1$ .
4. Извлекается 2 элемента очереди с минимальным приоритетом.
5. Создаётся вершина  $k$  с детьми  $i, j$ .
6. Суммируются два извлечённых элемента.
7. Добавляется в очередь сумма.

Написал программу (proccedurhaffman.py), которая реализует алгоритм, написанный выше

```
proccedurhaffman.py
prog > proccedurhaffman.py > ...
1
2
3 def Extractmin(H):
4     min = 100
5
6     for i in range(len(H)):
7         if min > H[i][1]:
8             min = H[i][1]
9
10    for i in range(len(H)):
11        if min == H[i][1]:
12            H.pop(i)
13            break
14    return min
15
16 def huffman(frequencies):
17     rez = []
18     H = []
19
20     for i in range(len(frequencies)):
21         H.append([i, frequencies[i]])
22
23     for k in range(len(frequencies), 2*len(frequencies)-1):
24         a = Extractmin(H)
25         b = Extractmin(H)
26         c = a + b
27         H.append([k, c])
28         rez.append([c, a, b])
29     return rez
30
31 if __name__ == "__main__":
32     frequencies = [3, 5, 10, 21]
33     print(huffman(frequencies))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\leo\Desktop\gitalg\lab7-alg> cd prog
PS C:\Users\leo\Desktop\gitalg\lab7-alg\prog> python3 .\proccedurhaffman.py
[[8, 3, 5], [18, 8, 10], [39, 18, 21]]
```

Рисунок 1 – Результат выполнения программы proccedurhaffman.py

Вывод: в результате проделанной работы было выяснено как строить дерево по коду хатфмана.