Практическое задание 2_3_2

Задание 1

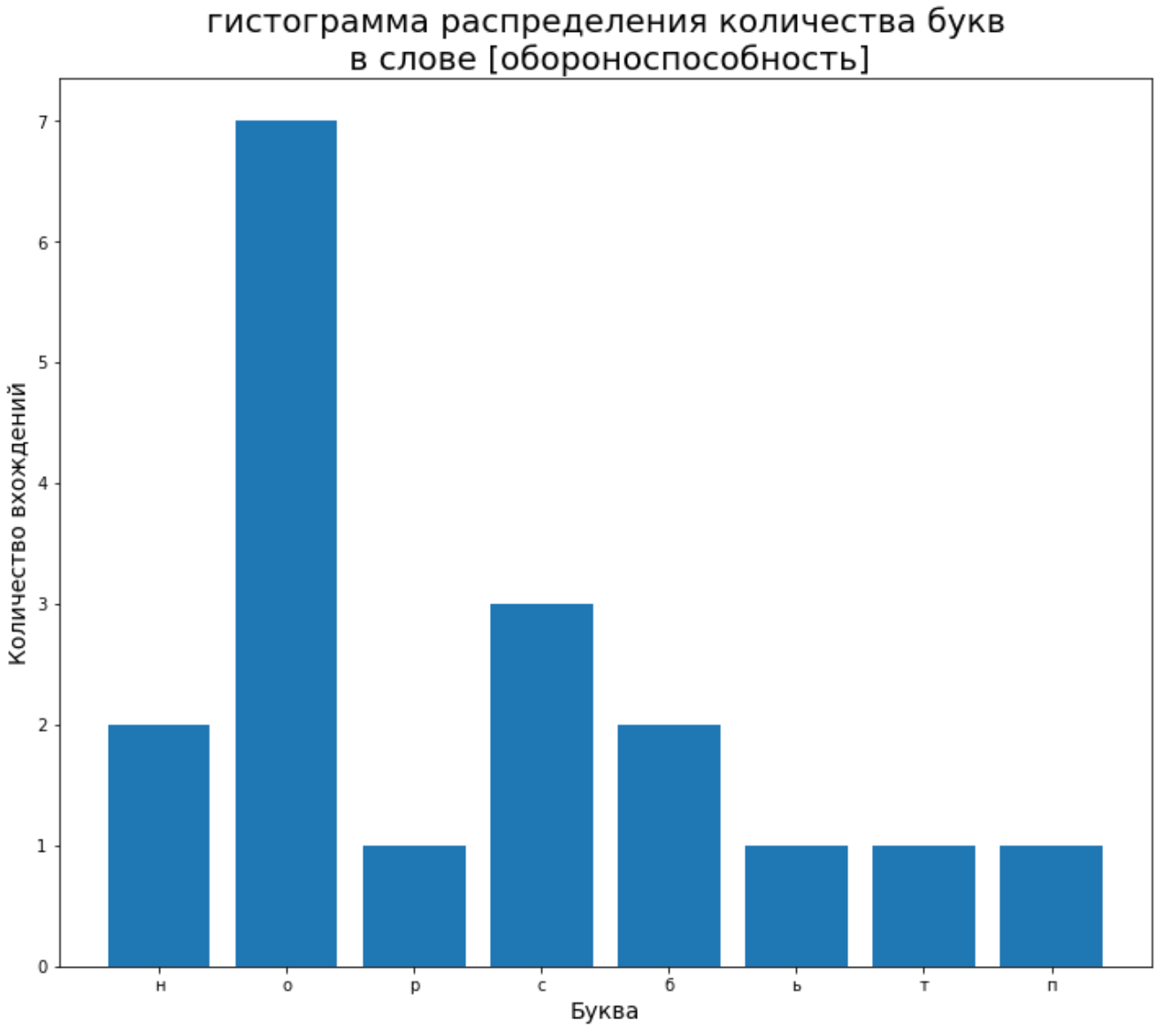
Какое слово написано? **нноороосссооообьбтп**

Напишите программный код для подсчёта количества вхождений каждой буквы в слово и постройте гистограмму числа вхождений букв. Сделайте подписи к рисунку и осям. В подпись к рисунку должно входить определенное Вами слово

In [ ]:

```
from collections import Counter
import matplotlib.pyplot as plt

word = "нноороосссооообьбтп"
counter = Counter(word)
x = list(counter.keys())
y = list(counter.values())
plt.figure(figsize=(12,10))
plt.ylabel("Количество вхождений", fontsize=14)
plt.xlabel("Буква", fontsize=14)
plt.bar(x, y)
plt.title("гистограмма распределения количества букв\n в слове [обороноспособность]", fon
tsize=20)
plt.show()
```



Задание 2

Разработайте приложение для поиска анаграмм в тексте. На входе у приложения - текстовый файл, на выходе - списки анаграмм

In [1]:

```
import random
import re
import nltk
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download('stopwords')
nltk.download('punkt')
stop_words = set(stopwords.words('russian'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

In [2]:

```
def anagram_finder(text):
    text = text.lower()
    word_token = text.translate(str.maketrans("", "", string.punctuation)).replace("—",
"")
    tokens = word_tokenize(word_token)
    result = [i for i in tokens if not i in stop_words]
    anagrams = []
    for  word_1 in  result:
        for word_2 in result:
            if word_1 != word_2 and (sorted(word_1)==sorted(word_2)):
                anagrams.append(word_1)
    return anagrams
```

In [7]:

```
#text = 'Аз есмь строка, живу я, мерой остр. За семь морей ростка я вижу рост. Я в мире -
сирота. Я в Риме - Ариост.'
f = open('text.txt', 'r')
text = f.read()

print(anagram_finder(text))
```

```
['есмь', 'строка', 'живу', 'мерой', 'остр', 'семь', 'морей', 'ростка', 'вижу', 'рост']
```

Задание 3

Разработайте приложение, которое принимает на вход текстовый файл (или просто текст) с обычным текстом и выводит этот же текст, но с перемешанными буквами внутри слов. Первая и последняя буквы каждого слова должны остаться на своих местах

In [ ]:

```
inputstring = '''По результатам исследований одного английского университета, не имеет зн
ачения, в каком порядке расположены буквы в слове. Главное, чтобы первая и последняя бук
вы были на месте. Остальные буквы могут следовать в полном
беспорядке, все равно текст читается без проблем. Причиной этого является то, что мы не ч
итаем каждую букву по отдельности, а все слово целиком'''
```

In [ ]:

```
def shuffle_string(inputstring):
    words_list = word_tokenize(inputstring)
    new_string = ""
    for i in words_list:
        shuffle_word = ""
```

```
        if len(i) < 2:
            new_string = new_string + i
        elif len(i) == 2:
            new_string = new_string + " " + i
        else:
            first = re.findall(r'^\w', i)
            last = re.findall(r'\w$', i)
            cut_word = i[1:-1]
            cut_word_list = list(cut_word)
            shuffle_word = shuffle_word + ' '.join(first) + ''.join(random.sample(cut_w
ord_list, len(cut_word_list))) + ''.join(last)
            new_string = new_string + ' ' + shuffle_word
    return new_string
```

In [ ]:

```
print(shuffle_string(inputstring))
```

```
 По ртзлеаьатум инидоласвсей огндоо акйгинсолго утнестевриа, не иемет зчниаеня,в каком п
дрякое рпеоасжлноы бвкуыв слвое. Гнлвоае, чтбоы прваеяи пялдосеня буквы были на мтсее. Оь
асытлне бквуы мугот сетодлваьв понлом бяксдрпоее, все равно ткест чтитсеая без полебрм. П
рниоочй эотго ятвесяля то, что мы не чеиатм куджаю бвуку по отнльдеотси,а все солво цклои
ем
```