Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Дисциплина «**Алгоритмы и структуры данных**»

ОТЧЁТ

по лабораторным работам за 9 неделю

Студент **Пастухов К.А. группы P3218**

Санкт-Петербург
2018 г.

Наивный поиск подкорки в строке

Даны строки p и t. Требуется найти все вхождения строки p в строку t в качестве подстроки.

Выполнение

```
import re
def get_indexes(string, sub_string):
    return [m.start(0) + 1 for m in re.finditer('(?='+sub_string+')', string)]
def main():
    with open("input.txt", "r") as file:
        sub_string = file.readline().strip()
        string = file.readline().strip()
    result = get_indexes(string, sub_string)
    with open("output.txt", "w") as res:
        res.write(str(len(result)) + '\n')
        res.write(' '.join(str(r) for r in result))


if __name__ == "__main__":
    main()
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.406	11087872	20003	48889
1	OK	0.046	9932800	14	6
2	OK	0.046	9867264	6	4
3	OK	0.046	9945088	6	3
4	OK	0.046	9965568	7	6
5	OK	0.046	9924608	7	3
6	OK	0.046	9981952	9	6
7	OK	0.062	9900032	10	4
8	OK	0.046	10080256	3004	3
9	OK	0.046	10031104	3028	6
10	OK	0.062	10047488	2656	428
11	OK	0.046	10117120	2005	8894
12	OK	0.062	10158080	4003	6
13	OK	0.062	10027008	3004	3
14	OK	0.046	9981952	2252	1849
15	OK	0.062	9957376	2021	185
16	OK	0.046	10149888	2008	8883
17	OK	0.062	10104832	3004	3903
18	OK	0.046	10010624	2670	3
19	OK	0.046	10080256	3028	6
20	OK	0.046	9928704	2404	690
21	OK	0.046	10207232	2005	8898
22	OK	0.062	10149888	4003	6
23	OK	0.062	9932800	2670	3
24	OK	0.046	9945088	2252	1885
25	OK	0.046	9928704	2022	189
26	OK	0.046	10145792	2008	8883
27	OK	0.046	10117120	3004	3903
28	OK	0.062	10080256	5337	3
29	OK	0.062	10153984	5028	7
30	OK	0.046	9945088	4372	647
31	OK	0.046	10350592	4005	18898

Карта
Выполнение

```
from edx_io import edx_io


def calculate_sum_distance(positions):
    k = len(positions) - 1

    sum = 0
    for p in reversed(positions):
        sum += k * p
        k -= 2

    for i in range(1, len(positions)):
        sum -= i
    return sum


def main():

    mapa = {}
    with edx_io() as io:
        tokens = [t.decode() for t in io.all_tokens()]
        sentence = ''
        for t in tokens:
            sentence += t
        for index, char in enumerate(sentence.replace(" ", "")):
            mapa.setdefault(char, []).append(index)
        sum = 0

        for _, val in mapa.items():
            amount = len(val)
            if amount > 1:
                sum += calculate_sum_distance(val)
        io.write(str(sum))


if __name__ == "__main__":
    main()
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.312	27537408	300002	16
1	OK	0.078	10993664	10	1
2	OK	0.078	11010048	34	3
3	OK	0.062	11046912	5	1
4	OK	0.062	11051008	6	1
5	OK	0.062	11091968	7	1
6	OK	0.046	11075584	9	2
7	OK	0.046	10977280	7	1
8	OK	0.046	10969088	7	1
9	OK	0.078	11059200	13	2
10	OK	0.046	10997760	202	6
11	OK	0.062	11083776	202	6
12	OK	0.078	10981376	202	6
13	OK	0.031	10948608	202	6
14	OK	0.062	11010048	202	5
15	OK	0.062	11042816	202	5
16	OK	0.093	10977280	202	5
17	OK	0.046	10997760	202	7
18	OK	0.062	11251712	5002	11
19	OK	0.062	11186176	5002	11
20	OK	0.078	11214848	5002	11
21	OK	0.062	11218944	5002	11
22	OK	0.062	11206656	5002	11
23	OK	0.062	11194368	5002	11
24	OK	0.062	11202560	5002	11
25	OK	0.062	11194368	5002	11
26	OK	0.062	11227136	5002	11
27	OK	0.046	11247616	5002	11
28	OK	0.062	11251712	5002	9
29	OK	0.078	11239424	5002	9
30	OK	0.078	11268096	5002	9