# Национальный исследовательский Университет ИТМО Мегафакультет информационных и трансляционных технологий Факультет инфокоммуникационных технологий

# Алгоритмы и структуры данных

Лабораторная работа №1

Работу выполнил: А.Э. Филиппов Группа: К3139 Преподаватель: Д. Добриборщ

 $ext{Санкт-} \Pi$ етербург2022

# Содержание

	1.1. Задача №3. Редакционное расстояние					
2.	Теоретическая информация					
3.	Ход выполнения работы					
	3.1. Список					
	3.2. Картинка					
	3.3. Таблицы					
	3.4. Листинг					

# 1. Задачи по варианту

#### 1.1. Задача №3. Редакционное расстояние

Редакционное расстояние между двумя строками — это минимальное количе ство операций (вставки, удаления и замены символов) для преобразования одной строки в другую. Это мера сходства двух строк. У редакционного расстояния есть применения, например, в вычислительной биологии, обработке текстов на есте ственном языке и проверке орфографии. Ваша цель в этой задаче — вычислить расстояние редактирования между двумя строками.

- Формат ввода / входного файла (input.txt). Каждая из двух строк ввода содержит строку, состоящую из строчных латинских букв. Длина обеих строк от 1 до 5000.
- **Формат вывода** / **выходного файла (output.txt).** Выведите расстояние ре дактирования между заданными двумя строками.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt	input.txt	output.txt
ab	0	shorts	3	editing	5
ab		ports		distance	

• Редакционное расстояние во втором примере равно 3:

l	$\mathbf{S}$	h	О	r	t	-
	-	p	О	r	t	$\mathbf{S}$

#### Код:

```
import time
     import tracemalloc
2
3
     tracemalloc.start()
     t_start = time.perf_counter()
     file = open('input.txt')
6
     lines = file.readlines()
7
     out = open("output.txt", "w")
9
     a = lines[0]
10
     b = lines[1]
11
12
     n, m = len(a), len(b)
13
     if n > m:
14
         a, b = b, a
15
         n, m = m, n
16
17
     curr row = []
18
     for i in range(n+1):
19
         curr row.append(i)
20
21
     for i in range(1, m + 1):
```

```
prev_row, curr_row = curr_row, [i] + [0] * n
23
         for j in range(1, n + 1):
24
             move_aus_a = prev_row[j] + 1
25
             move_aus_b = curr_row[j-1] + 1
26
             fortsetzen = prev_row[j-1]
27
             if a[j-1] \neq b[i-1]:
28
                  fortsetzen += 1
29
             curr_row[j] = min(move_aus_a, move_aus_b, fortsetzen)
30
31
     out.write(str(curr row[n]))
32
33
```

Для решения данной задачи использовалось расстояние Левенштайна. Вместо того, чтобы сохранять массив для ускорения работы сохраняются последняя строка и столбец.

#### Результат работы кода:



tqjopblrhzxjjrqfzifhyepilmigedfohgbrmdldkmyldnqihizxhjceesublpqvofwrykckrholujqialbuvyazobkzfbjnjrzxzayehxkxtwuvjjqsjnfsws qzaksjwpfimwmsdcxxzsdcxsacrvxhvdsezotwnnsfhzjnsrdmcuixjqabbvetplntglktkzskhowqinascfzqecjpbytmdvmalujyveokzybzhonvpjphnkd tqvygotepgmcgkcozpvbrcrfobkyjjchvksquyshslbhzermbkyoaiecfxhzmllkivyjkiaajtxakcjgxrjcyjwieqxipyybwhwbcasuvlgvnadsoiegoqbyot cbrtlrsnnplnbtmvnnvyjwomttgttlsmdhbewdllevyfeyqnadtgoaiotzniwktmkgarmdtmotyucunzbodovqcjkklbbghevrdnkpaodphubizvigksdyxcnz lfxfyvwukodmigqehtxfptuvzovvciwroznxuyphzgwpqwlfelgrzrvtkmuekrrfjbavjgyjbxyjmhjqlxjivfkvrphgwqfqbqdtazcgbsbrrdcwhtbfgsrzvc hrywbmyojhuauchhdlsszmynwheeqzrigqidlegndomyrbwcngjvskpicvaavyatsdfjazihcjybkbikzasgmrrwyvlipyngkbdgjzgfwxfgpfvcjmcopukwou hilsneuuhvooffxyiuyogedjvljprtjjplbyjpzekexkcmyuxqslcubydzlevuaxeqdkamjmqpfbgfrqeoxsjrymziatymsfhkwilnkdcjcmizmjbbzpaodwdt yzlutbxnwhljgtqhlnrxqeyrlhsfrlsrnpxdvsycmvfowj pnjcphaomermatzfuvhycesvgniqtckycqqfzvenpgvimpkumertxyjczyjohduiolwtgfizbfojnbfsjskcmbeqhbmskmoslwgrknejjdanledxtqpaidvasyxlxqrdrukhgtfdzkqurzihhekmoffeqaacwvksgfiskxzodbpcjbcfzpkruxzvxrpoqfnamyoav rtosfifulxtmauwswvfbwaoerejsuopbrsqkoivyswvijygcplbgngxnfxzbigvgpmwbuftxxsnbmopogoxplsancmdovlehfplijqilvxalisnolcmjxnozgy eyxonirnoorapbbfxdnwocobjmiebceqbnctacefpycabdnehrhkqsnlosipbpbxdqeuerasjovxfeurowqxanahmisffnnkximxoncfnkkpfxjhlnyastvbcubumtbspjtaxqnxqbegyrdvfzbxahiqdqtnijcdypdjttoipnnqmwclbahnzamkwtkqahmeofgkbtbuqelekpqyvnrlqjseibvbkfgutogtvtrhifjptpnwqahrubireebnbnjaeindjqwrznonpwosdwnobwowkrgsxyjwuhstxhawgqdwcznjeqlxqkzqytoozdxgedcpwxnwwjmiuepmckgnhqutskyhgdzyyviacevfsyrwacsvwrcgxhlxvhrsqfqapnfrzqkjlelhjjldrantiplmpxxepgcrzzlhlxdzqdpvombyxoopnpzojinvdokyltypkihdjmkolkczthegfkpenxanfbxphocwRMAL input.txt[+]

	Время выполнения (с)	Затраты памяти (байт)
Нижняя граница	0.00017152299915323965	13939
диапазона значений		
входных данных из		
текста задачи		
Пример из задачи 1	0.0001822340000217082	13941
Пример из задачи 2	0.00019643999985419214	13947
Пример из задачи 3	0.0002520400003049872	13952
Верхняя граница	1.9091916950001178	92904
диапазона значений		
входных данных из		
текста задачи		

**Вывод по задаче:** Расстояние Левенштайна считается самым эффективным алгоритмом для расчета редакционного расстояния. Его сложность -  $O(n^2)$ 

#### 1.2. Задача №4. Наибольшая общая подпоследовательность

Вычислить длину самой длинной общей подпоследовательности из двух по следовательностей. Даны две последовательности  $A=(a1,\,a2,\,...,\,an)$  и  $B=(b1,\,b2,\,...,\,bm)$ , найти длину их самой длинной общей подпоследовательности, т.е. наибольшее неотри цатеьное целое число р такое, что существуют индексы  $1 \boxtimes i1 < i2 < ... < ip <math>\boxtimes$  n и  $1 \boxtimes i1 < i2 < ... < ip <math>\boxtimes$  m такие, что  $i1 = bi1,\,...,\,aip = bip$ .

# 2. Теоретическая информация

bash [bash]

# 3. Ход выполнения работы

#### 3.1. Список

- первый элемент списка
- второй элемент списка

#### 3.2. Картинка



Рисунок 3.1. название картинки

Текст без отступа (следует за вставкой) Новый параграф Новый параграф с принудительно выключенным отступом

# 3.3. Таблицы

#### Одна таблица

 Element
 First
 Second

 One

 Two

 Three

 Four

Таблица 3.2

Таблица 3.1

# Другая таблица

	top center	
bot left	bot center	bot right

#### 3.4. Листинг

```
#!/bin/bash
1
2
     export LANG=en_US.UTF-8
^{3}
     dir_c=-1
5
     file_c=0
6
     cool() {
              dir_c=$(($dir_c + 1))
9
              local directory=$1
10
              local prefix=$2
11
12
              local leaves=($directory/*)
13
              local local_c=${#leaves[@]}
14
15
              for i in "${!leaves[@]}"; do
16
                      local leaf=${leaves[$i]}
17
                      leaf=${leaf##*/}
18
19
                      local local_prefix="
20
                      local pointer=" - "
21
                      if [ $i -eq $((local_c - 1)) ]; then
23
                               pointer="└─ "
24
                               local_prefix="
25
                      fi
26
27
                      echo "${prefix}${pointer}$leaf"
28
                      [ -d "$directory/$leaf" ] & cool "$directory/$leaf"
29
         "${prefix}$local_prefix" || file_c=$((file_c + 1))
30
              done
31
     }
32
```

Листинг 1: Script.bash – bash в массы!

### Заключение

IATEX удобен для создания отчётов, так как сам следит за нумерацией таблиц, рисунков, листингов и отсылок к ним (так, например, здесь всегда будет указан номер рисунка "sample" не зависимо от того, какой он (1,2 или другой) - это рисунок 3.1). Не менее важно что весь документ оформлен в едином стиле, а исходные материалы подключаются к отчёту, а не хранятся в нём. Всё это позволяет легко получить качественный отчёт без дополнительных трат на его офрмление.

Исключения, пожалуй, составляют таблицы, так как их значительно сложнее создавать кодом, нежели в графическом редакторе. Но здесь никто не запрещает использовать визуальные средства создания таблиц для  $\LaTeX$ .