Анализ Алгоритмов

семинары, л/р: Волкова Лилия Леонидовна

л/р: Строганов Юрий Владимирович

Семестр: мод. 1: 4 л/р + РК $\rightarrow$ в срок 4 лабы автом. РК

мод. 2: 4 л/р + РК

мод. 3: зачёт по теор. бил.

1. Лаба: Расстояние Левенштейна, расстояние Дамерау-Левенштейна (рекур. выч-я)

Пример рек. формы - числа Фибоначчи

1) Тривиальные услов: $f_0 = 1$, $f_1 = ?$.

2. $k_i > 2$; $f_i = f_{i-1} + f_{i-2}$

Расстояние для чисел - разница

для 2D, 3D - точек - эвклидов. расст.

для строк - редакционное.

расстояние, или расстояние Левенштейна

Пусть требуется преобразовать строку s1 в строку s2.

Редакторские операции:

1) вставка символа в s1 (I, insert) } штраф 0

2) удаление символа из s1 (D, delete) }

3) замена символа (R, replace)

4) совпадение символами s1 и символ s2 } штраф 0

Расстояние Левенштейна - минимальное необходимое для преобразования строки s1 в строку s2.

Кол-во редакторских операций.

S_1: р а з в л е ч е н и я

S_2: у в л е ч е н и е

Операд   R R R R M R R R D

[0 - это не ~~... mln~~ min расстояние.

S_1: раз в л е ч е н и я

S_2: у в л е ч е н и е

P D R M M M M M M R

min расст - ч.

Чтобы проблему сопоставления слов по слож. фла-
там, рассчёт ведут там, по формуле:

Обозначим $D(S_1, S_2) = D(S_1[1 .. \underset{\text{длина}}{L(S_1)}], S_2[\underbrace{1 .. L(S_2)}_{\text{подстрока}}])$

$$D(S_1[1..i], S_2[1..j]) =$$

$$\min \begin{cases} D(S_1[1..i], S_2[1..j-1]) + r, & // I \\ D(S_1[1..i-1], S_2[1..j]) + r, & // D \\ D(S_1[1..i-1], S_2[1..j-1]) \end{cases} + \begin{cases} 0, & \text{если } S_1[i] \\ 1, & \text{иначе } // R \end{cases}$$

Тривиальный случай:

$$D(\varnothing; \varnothing) = 0, \quad D(\text{я}, \varnothing) = 1 \quad // D$$

$$D(\varnothing, \text{я}) = 1 \quad // I$$

Реализация через рекурсию:

D ("скат", "кот"

D ("скат", "ко") + 1 D ("ска", "кот") + 1

$/D$ ("ска", "ко") + 0
т.к. "т" = "т"

D ("скат", "к") + 1 (D ("ска", "ко") + 1 D ("ска", "к") + 1
т.к. "т" ≠ 0.

D ("скат", ∅) + 1   /D ("ска", "к") + 1

}
4

}
Много-однотипных
выз-в.

Реализация ~~корреляция~~

с помощью матрицы $(L(s_1 + 1)) \times (L(s_2) + 1)$

1) Тривиальный случай: 0-я строка, 0-й столб.

В. ∀ клетке
D ($s_1[1..i]$, $s_2[1..j]$).

расстояние между подстрокой s_1
длиной i и подстрокой s_2 длиной j

2) для $i > 0$, $j > 0$ по формуле
$$D("c","k") = \min \begin{cases} D("c", ∅) + 1, \\ D(∅, "k") + 1, \\ D(∅, ∅) + 1 \end{cases}$$
т.к. буква разн.

! штраф, может быть 0, только по
диагонали.

Сравниваем i-й символ S_1 ('i') и j-й символ s_2. Они разные ⇒ штраф по диагонали 1.

Если представить, что матрица - это карта вы[сот], то маршрут спуска из правого нижнего угла матрицы в лев. верхний даст ну послед. редакторских операций (возможно, не единств.) кот. привела к отв[ету] (в обрат. порядке).

Применение 1: <u>поисковых сист-м</u> при некоррект. Вводе должн найти словарное слово, наиболее близкое к тему, что. в ведро

Применение 2.: <u>биоинформатика.</u>

Можно шифровать белки строкой из букв, каждая из которых обозначает молекулу.

Расстояние Дамерау-Левенштейна:
Одна из самых частых ошибок текста, это перестановка 2 соседних букв.
Вводится операция перестановки
(транспозиции)

В случае если $i>1$ и $j>1$ и $s_1[i]==s_2[j-1]$ и и $s_1[i-1]==s_2[j]$, в минимум, войдёт 4я операц[ия]
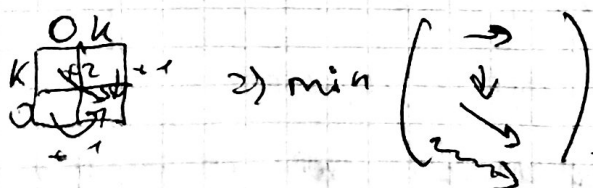
Пр ① Практич. часть - реализ. матриц расст.
Левенштейна матрично и рекурсивно. р. Дамерауд.

Ввод: 2 строки
②вых: расст. и матрица (для не риуса)

③ Замер времени и показать.
    рассчёта расст. (процессорное время)

    сколько t процессор потратил
    именно на эту задачу.

+ Отчёт в эл. форме. Процедура, та же, что РПЗ

Титул (ФИО, группа, название л/р, год)

Оглавление
Введение
I Аналит. часть
II Конструкт. часть.
III Технологич. часть.
IV Эксперимальная часть (исслед.).

Заключ.

② разр.