Fachbereich Informatik

Prof. Dr. Nikolaus Augsten
Jakob-Haringer-Str. 2
5020 Salzburg, Austria
Telefon: +43 662 8044 6347
E-Mail: nikolaus.augsten@plus.ac.at

PARIS LODRON UNIVERSITÄT SALZBURG

Similarity Search in Large Databases — Prüfung
Wintersemester 2022/2023 — 30.01.2023

Name: ______________________ **Matrikelnummer:** ______________

Hinweise

- Bitte überprüfen Sie die Vollständigkeit des Prüfungsbogens (9 nummerierte Seiten).
- Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf jedes Blatt des Prüfungsbogens und geben Sie alle Blätter ab.
- Grundsätzlich sollten Sie alle Antworten auf den Prüfungsbogen schreiben.
- Sollten Sie mehr Platz für eine Antwort benötigen, bitte einen klaren Verweis neben die Frage auf die Seitennummer des zusätzlichen Blattes setzen.
- Keinen Bleistift verwenden. Keinen roten Stift verwenden.
- Verwenden Sie die Notation und die Lösungsansätze, die während der Vorlesung besprochen wurden.
- Aufgaben mit mehr als einer Lösung werden nicht bewertet.
- Als Unterlage ist ein beliebig (auch beidseitig) beschriftetes A4-Blatt erlaubt.
- Zeit für die Prüfung: 90 Minuten

Unterschrift ______________________

Korrekturabschnitt — Bitte frei lassen

Aufgabe	1	2	3	4	5	6	7	8	Summe
Maximale Punkte	2	2	2	2	2	2	2	2	16
Erreichte Punkte									

Exercise 1 - *Sorted Neighborhood Matching*. 2 Points

Consider the clean-clean entity resolution problem in Figure 1. Use the sorted neighborhood technique to produce *candidate pairs*, i.e., the record pairs that must be compared. Sort lexicographically in ascending order by *Name*, *PLZ*, and *YoB*, and use a window size of $w = 2$.

Figure 1: Clean-clean entity resolution problem.

ID	Name	ZIP	YoB
a_1	Smith	5020	1993
a_2	Christen	5020	1998
a_3	Huber	5034	1993
a_4	Buber	5034	1949
a_5	Gruber	5010	2011

ID	Name	ZIP	YoB
b_1	Gruber	5034	1998
b_2	Smyth	5020	1993
b_3	Huber	5034	1949
b_4	Gruber	5020	2011
b_5	Chirsten	5020	1998
b_6	Huber	5034	1993

Exercise 2 - *String Edit Distance.* 2 Points

Compute the edit distance between the strings `proof` and `porof`. Use the matrix produced by the dynamic programming algorithm to derive the shortest edit scripts and represent them with the gap representation.

Exercise 3 - *q-Gram Distance*. 2 Points

Given the strings x = `blabla` and y = `alaba`. Compute the q-gram distance and the normalized q-gram distance between x and y ($q = 2$).

Exercise 4 - *Opitmal Substructure of the String Edit Distance Problem.* 2 Points

Proof the optimal substructure property of the string edit distance problem:

> Given a gap representation, $\text{gap}(x, y)$, between two strings x and y, such that the cost of $\text{gap}(x, y)$ is the string edit distance $\text{ed}(x, y)$. If we remove the last column of $\text{gap}(x, y)$, then the gap representation of the remaining columns, $\text{gap}(x', y')$, has cost $\text{ed}(x', y')$ between the resulting substrings, x' and y'.

Exercise 5 - *Forest Distance Matrix.* 2 Points

Consider ordered trees T_1 and T_2 in Figure 2, forest distance matrix fd, and tree distance matrix td for the trees T_1 and T_2.

a) Compute the left-most leaf descendant arrays l_1, l_2 for trees T_1, T_2, respectively.

b) Fill the missing values for d_i and d_j into the forest distance matrix fd for the circled keyroot nodes in trees T_1 and T_2.

c) Circle the cell in the forest distance matrix that stores the distance between the prefixes $T_1[3..4]$ and $T_2[1..3]$.

d) Cross all cells in the forest distance matrix and/or the tree distance matrix required to compute the distance between the prefixes $T_1[3..4]$ and $T_2[1..3]$.

fd	$\vec{d_j}$					
$d_i \downarrow$						

td	1	2	3	4	5
1					
2					
3					
4					
5					
6					

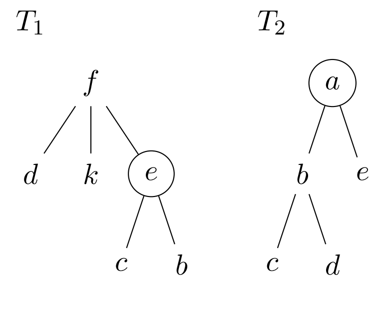

Figure 2: Two ordered trees T_1 and T_2.

Exercise 6 - *Constrained Edit Distance.* 2 Points

Consider ordered trees T_1 and T_2 in Figure 3. Show an edit mapping between T_1 and T_2 that is *not* a valid constrained edit mapping. Explain.

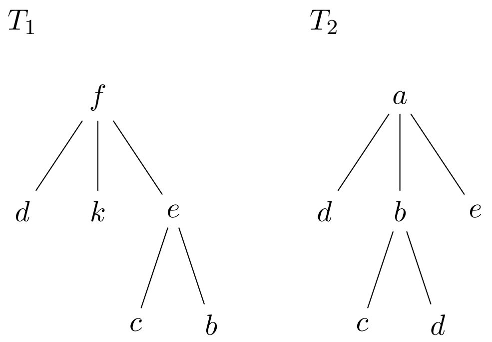

Figure 3: Two ordered trees T_1 and T_2.

Exercise 7 - *Binary Branch Distance and Lower Bound.* 2 Points

For the ordered trees T_1 and T_2 in Figure 4:

a) Represent T_1 and T_2 as normalized binary trees and compute the binary branch distance.

b) Based on the binary branch distance, what is the smallest value that the edit distance between T_1 and T_2 can adopt?

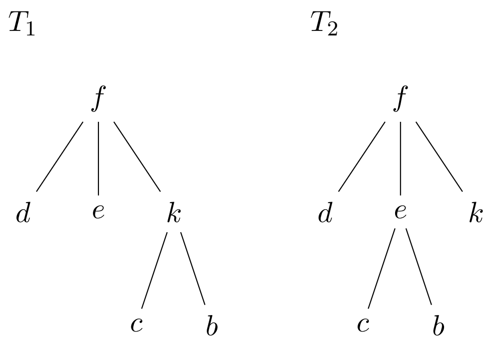

Figure 4: Two ordered trees T_1 and T_2.

Exercise 8 - *Cosine Prefix Signature.* 2 Points

Consider the collection $R = \{s_1, s_2, s_3, s_4\}$ of sets in Figure 5. Compute *prefix signatures* for all sets $s_i \in R$ for *cosine similarity* threshold $t = 0.9$.

Note: For the cosine similarity, $Cos(r, s)$, between two sets, r and s, the following holds:

$$Cos(r, s) \geq t \Rightarrow |r \cap s| \geq t^2|r|$$

$$\begin{aligned}
s_1 &= \{A, C, B, D, F, E\} \\
s_2 &= \{D, E, F, B, A\} \\
s_3 &= \{B, D, F, G\} \\
s_4 &= \{C, B, G\}
\end{aligned}$$

Figure 5: Set collection $R = \{s_1, s_2, s_3, s_4\}$.