

Prof. Dr. Goran Glavaš,
M.Sc. Fabian David Schmidt
M.Sc. Benedikt Ebing

Lecture Chair XII for Natural Language Processing, Universität Würzburg

1. Exercise for “Algorithmen, KI & Data Science 1”

15.12.2023

1 Dynamic Programming

1. Recall Merge-sort from lecture 5. Would it be beneficial if we use top-down dynamic programming with memoization in merge-sort? Explain briefly why or why not.
2. We want to compute the Levenshtein edit distance between *Frodo* and *Gondor*. Consider the sub-problem of computing the distance between *G* and *Frod*. What are the costs for insertion, deletion and replacement at this stage, respectively.
3. Write down the full 6×5 array of distances between all prefixes as shown in the lecture 13. What is the minimum edit distance between Frodo and Gondor?
4. Implement `levenshtein_distance` in the accompanied Jupyter Notebook using memoization.
5. Consider the following two-player game: There is a row of n objects (assuming n is even) with values v_1, v_2, \dots, v_n . In the game, the two players make moves alternatively. In each odd move, the first player either selects the first or the last object of the row and removes it permanently from the row. In even turns, the opponent plays the game in the same way. Each time the player picks some object, it receives its value.

Devise a dynamic programming algorithm akin to the four steps as per the lecture to determine the maximum value that the first player (i.e., the player starting the game) can receive by the end of the game.