Project **Winter 2021/22**
Algorithm Engineering for hard problems **TU Berlin**
Nichterlein/Boehmer/Niedermeier **November 3, 2021**

2nd Exercise Sheet

Weighted Cluster Editing I

Solver: November 22 **Handout:** November 23 **Presentation:** November 24

Task 1. Improved Search Tree with Lower Bounds

In the 2nd screencast we presented an improved search tree algorithm (called edge branching or merge branching) with search tree size $O(2.619^k)$. Implement the search tree algorithm with at least two different lower bounds variants to reduce the number of recursive steps. Ensure that your program's input and output work with the provided benchmark script.

Task 2. Evaluation

Test your program from on the provided instances using the benchmark script. Count the number of recursive steps by adding in the output the line "`#recursive steps:` x", where x are the number of recursive steps. The benchmark script will pass this information to the console.

Hint: Further requirements on the evaluation are given in the next task.

Task 3. Presentation & Handout

Please submit a handout of your evaluation and present your findings on the next day with a short talk lasting at most 10 minutes.

The main focus of the handout should be

- the evaluation and comparison of your implemented lower bounds (does not apply to two-person teams),
- the theoretical and the observed practical running times (depending on k and n) and search tree size[1], and
- the comparison of your submitted solvers from this and the last exercise.

Differentiate between the instance types *random*, *real-world*, and *actionseq*. In the talk, come up with possible explanations for observed differences.

You have a space limit of one page of text in the handout plus one page of diagrams (all need to be referenced and discussed in the text).

Hints: You might want to adapt the benchmark script to test all these variants of your program at once. If you do not know how to adjust it, ask in the forum. We will then explain what to change.

[1] You might want to use the number of recursive steps here.