

author = “8168265, Karabacak, 8175858, Braun”

Analysis:

The program is developed according to the EVA design pattern.

Input and output are done in the console.

The program goes through each matrix which are stored in a list via a for loop.

The two methods `greedy_traversal` and `find_best_way` from `ue_07.py` module are tested.

Output: Output is done in the console.

Description of the program:

The program does not require any further libraries other than `timeit` and `copy`, it needs the `ue_07.py`.

The program is written and tested in Python 3.9.6. A Python interpreter must be installed.

The program can be started with the command “`python3 ue07.py`” in the terminal.

The program uses `timeit` to measure the time it takes to run the functions. We measure by executing the function 10 times, so that the result is more readable and passes through bigger matrices in a more relatable time .

Test results:

Look via Excel file named `epr_07_v1_ex3.xlsx`

Comparison of time and costs `greedy_traversal(greedy)` vs `find_best_way(optimal)`

The greedy algorithm is worthwhile for the (2x2) matrix because, due to the small matrix, the way both greedy and optimal are taking are the same. The time, on the other hand, is always better for greedy because it only goes through the (2x2) matrix once. Between the (2x2) matrix and the (6x6) matrix, it is more worthwhile due to the costs and the small time difference. Only from the (6x6) matrix onwards does greedy become worthwhile again, since the time of the optimal function takes significantly more time to go through the entire matrix, whereas with greedy the time is still as short as with the (2x2) matrix is, i.e. from there you should take on more costs so as not to waste too much time.