Problem1:

I firstly define the file name and set the first input argument as variable N. Then I read the json file using from_caliper. Afterwards I sort the dataframe by time in decreasing order and take the first N lines, so I get the N functions which spend the longest time. Then I select the columns 'name' and 'time' and use reset_index(drop=True) to drop the default index in the dataframe. Finally I also drop the index and header so that the output to stdout will be function name and time taken only.

The most important function here is sort_values() and head(), which in combination could select the top N lines in favored order.

Problem2:

As in problem1, I firstly read the json file and set input argument to be X. Then I use load_imbalance(verbose=True) to get the imbalance graph frame. Here verbose should be 'True', or the wanted attribute 'time.rank' will not be provided. Then I change the graph frame to dataframe and sort it by 'time.imbalance' in decreasing order and take head(X). This means I get the top X nodes with the longest imbalance time. Then I select the last node from the X nodes using tail(1), so that this node will be the one with Xth longest imbalance time. Finally I drop index and header and output the cololum "time.ranks" only to stdout as the list of processors that have Xth longest imbalance.

The most important function here is load_imbalance(verbose=True). Without it I can never get information about imbalance time and 'time.ranks'.

Problem3:

This time I read two files and store in two different graph frames. Then set the first input to be N. Then use drop_index_levels on these two graph frames respectively to drop the additional index columns in the hierarchical index. This makes the two graph frames easier for comparison. Then I subtract the 8-core frame from the 64-core frame and turn the difference into a dataframe. Again sort it by time in decreasing order and take the top N lines. Then select the "name" and "time" columns only by dropping index and header.

The most important function is that we can directly get the difference by subtraction, i.e., diff = gf64 – gf8.