Short overview of the code

1. Import GUROBI
2. Create a function called ‘min_distance’, the input is number of teams followed by a distance matrix for each instance (txt.).
3. Extract number of clubs, number of rounds from the data and create sets for all the teams and rounds. Create variable, distance matrix Dij.
4. Create MIP model decision variables based on the mathematical model in final report.

   More specific:

   - X_vars =X_{ip}: if team i plays home game in round p; 1 for yes, 0 otherwise
   - Z_vars =Z_{ijp}: if team i and team j play game together in round p; 1 for yes, 0 otherwise
   - C_vars =C_{ip}: travel distance for team i between round p and round p+1
   - L= L total travel distance (our objective is to minimize it)
5. Set MIP constraints based on model constraints in final report.

   The order of constraints in Python script are in consistency with the order of constraints in model overview from the final report.
6. Use GUROBI optimizer (minimize objective function) to get the optimal solution of the MIP model.
7. Get optimal tournament schedule. The first number in output is the minimum total travel distances. The following dictionary indicates which two teams play together in each round, and the team in the first place of a bracket is the team that plays home game.