Task A

For the first task we created a dictionary. The keys of dictionary are the blocking keys such as ‘inc’, ‘qd’ and for the values we used a list. So, each key contains the list of IDs where we can find this particular word. After that we created the blocks and we printed the keys with their corresponding values (authors, venue, years, title). We also created a function that asks for an input, which input should be one of the blocking keys such as ‘jhw’ and prints the attributes that are found inside that block.

Task B

For the second task we simply calculated the number of comparisons that have to be made for each block and summed them. As number of comparisons for each block we used the following formula:

n*(n-1)/2

We found 1911112840 total comparisons that have to be made.

Task C

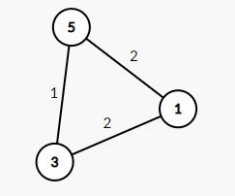
For the third task we took a sample of blocks. We did this because my pc could not handle this volume of computations. We took 10 blocks from those that were created in Task A and found the initial number of comparisons that had to be made, which are 1539826. So, for these 10 blocks if we were to make any comparison, we will have to make 1539826 comparisons. After that, we created a Graph with each node representing an ID from the data frame and each edge representing the sum of occurrences of these IDs in the blocks and later we pruned the edges for weight < 2. For example, assume these 3 initial blocks with their IDs:

Block A: {1, 5}

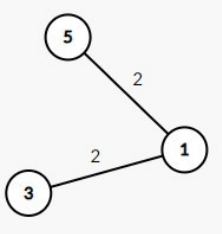
Block B: {1, 5, 3}

Block C: {1, 3}

For these blocks we create the following initial Graph:



Now we Prune for weight < 2 and we are left with the following Graph:



So the new blocks are: {1,3} and {1,5}

In our case, after the creation and pruning of the Graph we found 104356 number of comparisons that had to be made. Also, we have created an undirected Graph. So, the edges that are retained after the pruning create blocks of minimum size (that’s 2) with each block containing 2 IDs that are connected through the edges of the Graph.

Task D

For the final task we simply created a function that calculates the Jaccard similarity for two entities only for the title attribute. This function takes as input two IDs and outputs their Jaccard similarity.