SUMMARY

The algorithm is divided into two python files mapper.py and reducer.py.

In the mapper.py python file, it reads the input Posts.xml file line by line. At the beginning of the loop, we create an empty list “index” and a flag variable (the purpose of using a flag is to know whether a line contains view count or not, if it does not contain then viewcount (key) is set to 0 and other two values as 1 and “empty” ). In the first for loop, we are appending the indices of double inverted commas to extract each element. Then in the next for loop, we are taking a pair of indices and checking that whether slicing the line in that range of index contains “Viewcount” or not, if it contains “Viewcount” then we will select the next two pairs of indices to extract the value of “Viewcount” and set flag as True, and if in the next iteration of the loop we found “Title” then we extract the value of “Title” and set as second value. For piping between mapper and reducer, we had printed the key and values pairs separated by single tabs, before doing this we removed the unnecessary characters from the key using the strip command. here the key is ‘viewcount’ and values are 1 and its title, and if no ‘viewcount’ is found in any particular line then the key is 0 and values are 1 and “Empty”.

In the reducer.py file, we used sys module and prettytable module to plot the table in a systematic order. Before reading the outputs from the mapper we will initialize some variables and created two empty lists, one to hold the top 10 viewcount and another for their title where their index will be the same. Then we will read the lines and split them based on the tab and store the values in the word, count, and x variables. And on the following code, we counted the number of posts that were viewed a given number of times and found the top 10 viewcounts. In the end, we created a table with columns “ViewCount” and “Title” and plotted the table.