COL362 - Assignment 3 - Report

Rishita Agrawal (2020CS50439)
Rajat Bhardwaj (2020CS50436)

Code Description:

MAXSIZE variable is the estimated block size, considering 1 GB memory and (k+1) blocks.
MAXSIZE in our case is = 90,00,00,000/1024

We are reading the input file using getline and a while loop, here we also take care about the key_count argument, (Note that if key_count is less than equal to 0, we will sort the whole text file).
These strings are being stored in a vector.

While reading the file if the vector size is become equal to MAXSIZE*(k+1), we sort the vector, create intermediate files, and empty the vector to refill it again. Thus, creating initial runs.

We are sorting using an inbuilt sort function for vectors.

Once these runs are created, we start merging. We use a loop to create merge passes. In each pass we merge the n files present, merging k files at a time.

(n is the number of files to be merged in one pass)

Since we can't read the whole file at once while merging, we just read MAXSIZE number of strings from each file at a time.We start merging them and writing the output on a buffer. If any memory block becomes empty we again refill that block more strings from the current run (temp file corresponding to which the block has become empty). If the buffer size become equal to the MAXSIZE we just dump the buffer to the output file and empty the buffer. We are using an array of MAXSIZE vectors, where each vector has maximum k strings. We maintain the head and tail index as we can not have an array of variable size.

Data Structures Used:

For sorting, we used a vector of strings.
We are using vectors majorly, but for merge passes we changed it to array of vectors instead of vector of vectors for optimisation.

I/O optimization:

For writing into the files, we are using a buffer, so that we can write to the file in single go instead of using a while loop for writing from a vector.

Other optimization:

For storing the chunks in blocks we have used array of vectors of strings. Each array is MAXSIZE long and each vector inside the array is of k size. We have used an index array and a maxsize array also because to iterate inside a block we need to iterate on the outer array with a fixed index of the inner array.

Also please note that the term ‘block’ is used just for the sake of understanding and it does not represent the actual memory block in the computer memory.

Testing:

We used a synthetically generated 10 gb file for checking time limit.
We were not able to upload it online as the file was too large.

The time came out to be approximately 120 sec. Out of which the sorting part took around 45 sec and the merging part had 60-70 sec, the rest was file read and write and other small operations.

Time for given test workloads:
English.txt - around 0.9 sec
Random.txt - around 4.5 sec