Description of Program

In this project our group modify 3 class from 7Zip follow from instruction in my course. The first class we did it is index class. The first thing that we did it on this class is delete output directory prepare for another step. The second thing that we did it is generate the word in the file that have in dataset to term ID and Doc ID in every single block. The third thing that we did is manage Term ID to have list of Doc ID and collect to posting List. After the third step, we store this posting to the file, we pick term ID, frequency and starting point of each term ID that have Doc ID to Posting File. The last step of index class is merge. We follow the extra class that requirement us to use BSBI to merge. We use BSBI to mix the two block together. On the Basic Index Class. We apply the algorithm from the textbook and internet together to write and read channel file. And the last class is Query class. On this class this class will load the corpus that contain the value of the merge by BSBI from Indexing. After this, this class need the token or single word to check or find. This class will load posting list from the file that we written in index class. After this is the step before the last step that is combine the document that contain the token or the word that user wants to check together. Moreover, our team consider how to get all document ID together, so we use link list to save the document ID because it can sort and easy to save more Document ID. And the last step is write the Document ID to the output file.

a) We asked you to use each sub-directory as a block and build index for one block at a time. Can you discuss the trade-off of different sizes of blocks? Is there a general strategy when we are working with limited memory but want to minimize indexing time?

Ans We always clear the process and other in memory because we prepare for new block to processing. Moreover, If we have limit memory we think we will decrease the loop and use BSBI algorithm and divide every part in index to make it process slow but use less memory.

b) Is there a part of your indexing program that limits its scalability to larger datasets? Describe all the other parts of the indexing process that you can optimize for indexing time/scalability and retrieval time.

Ans We always clear the memory in every write and read process. Moreover, on the merge part we keep it in link before write so it take less memory.

c) Any more ideas of how to improve indexing or retrieval performance? Please note if you go drastically beyond the 2-page limit, we may penalize you for an overly long report.

Ans We try to use loop less as much as we can and try to comment the printf that we use to check. Moreover, in the future search engine should use another algorithm.