Exercise Sheet 5
complete until Monday, November 28th, 9am

Exercise 1
Implement a compression scheme of your choice to improve the running time of the (simple, linear-time) list intersection you implemented for Exercise Sheet 2. That is, try to achieve a better trade-off between the time taken to read the inverted lists from disk, and the time taken for the actual intersection. Proceed in two steps.

Step 1: For each of your 26 lists (one for each letter of the alphabet), create a binary file that stores the document ids in ascending order, using 4 bytes per id. See the Wiki for a simple perl script that creates these files from the files which you (probably) produced for Exercise Sheet 2.

Step 2: Now compress these binary files with the compression scheme of your choice, and let your intersection algorithm decompress them before use.

Remark: there are two main options for implementing Step 2. For a simple compression scheme, one option is to implement the compression and the decompression yourself. When you do that, it will be easy for you to integrate it with your intersection algorithm. The other option is to make use of an external program for compression. The compression part is then obvious. For the decompression you then have to find a way to communicate with the external program, e.g., via a pipe. Check out the Wiki for hints and post a question if you need help.

Having done parts 1 and 2, you should post four numbers on the Wiki: the intersection rate and the disk read rate without compression (using the files from part 1) and these two rates with compression. As usual, also post the programming language and machine specifications.

Make an effort to improve over what you already achieved for Exercise Sheet 2, that is, the average of the two rates should become better.

Exercise 2
Let the compression ratio of a scheme be defined as the size of the compressed data divided by the size of the uncompressed data. Measure the ratio you achieved for Exercise 1, and post it on the Wiki. Then try to give a theoretical explanation for the ratio you observed. You need not (and probably cannot) derive the exact value, but rather explain its magnitude, that is, give theoretical evidence, why, on your data and with your scheme, you obtained a ratio of say 50% and not 25%. Try to be as precise and concise as you can, that is, do not just say “there is lots of repetitions in the bytes of my document ids, and my scheme compressed them”.