

**Illinois Institute of Technology**  
**CS429: Introduction to Information Retrieval**  
**Fall 2005**

Project Part 3: Sort-Based Inversion & Compression

**Date Given:** October 20<sup>th</sup>

**Due Date:** **November 17<sup>th</sup>** by no later than 3:00 pm on blackboard digital box & also a hard copy of all deliverables but the code.

---

**Grading:** This assignment is 15 points out of the total 35 points allocated for all assignments in the semester. It will be graded on the scale of 100.

**NOTE:** *You can use IRIS software or use any programming language to implement this assignment.*

**Objective**

This phase is to implement an efficient and scalable IR System. You can no longer assume that you have sufficient memory to store entire inverted index. You have to use the sort-based algorithm for building the inverted index. To reduce both the amount of I/O in query processing and the space requirements, you need to compress the inverted index. A suitable compression technique is to be used to achieve compression ratio of almost 1:3.

**Requirements**

Start with your completed part 2. Read the memory constraint parameter from the command line. This parameter specifies the memory requirements in term of number of triples, i.e, amount of data can be kept in memory. Use sort-based algorithm, taught in the class, to create inverted index. Capture system time needed to make the inverted index for each of the cases of memory constraint parameters. You are given four memory constraint parameters, i.e. 1000, 10000, 100000, and UNLIMITED triples. If the size of our triple lists after processing each document is greater than the size of memory constraint, then you should make memory available before processing to the next document by writing the triples onto the disk. For each of the memory parameter experiments listed above, capture maximum memory used just before writing the last triples list onto the disk.

Assumptions are:



1. Posting list of a given term fits into memory.
2. Relevance ranking scores obtained for top 100 documents for all 26 queries can be kept in memory.
3. Distinct term map for each document fits into memory.
4. Lexicon fits into memory.

After successful creation of inverted index with limited memory and gathering the statistics for report 1 (see below), use the compression techniques to compress the inverted index. Note the indexing time and the size of inverted index on the disk. Use a compression technique as listed below to implement the compression. Choose the compression technique that would give you a better compression ratio.

1. Byte Aligned
2. Elias encoding

**Runs & Reports:**

**Report 1:** Generate this report before you compress the inverted index. Execute queries using the similarity measure that gave you the best average precision for project 2. For each of four triples list sizes (1000, 10000, 100000, and unlimited) provide the following info:

Triple list size	1000	10000	100000	Unlimited
<b>Statistics</b>				
Time taken to build Inverted Index in milliseconds (the whole process) (C)				
Maximum Memory used before writing the last triples list onto the disk in bytes				
Index size in bytes. <del>This includes triple list, lexicon, and document name to document id mapping (document map).</del> (A)				
Query processing time (over 26 queries) in milliseconds. (From time getting queries till producing results, i.e. ranked scores. Do not time the results generated by treceval) (E)				
Interpolated average precision determined by treceval using cosine  (attach treceval report) (G)				

Note: submit the treceval summary reports, each in a separate page (4 summary reports).

**Report 2:** Implement compression now and give following values in the table provided for triple list size of 100000. Execute queries using the similarity measure that gave you the best average precision for project 2.

Compressed Index size (B). <del>Include all components of index as in (A).</del> in bytes	Compression ratio (B/A)	Indexing time in milliseconds (D)	Indexing time ratio (D/C)	Q. proc. time over 26 queries in milliseconds (F)	Q. proc. time ratio (F/E)	Avg precision (H)	Avg precision ratio (H/G)

Note: submit the treveal summary report (1 summary report).

### Deliverables

1. Summary (10%): include the following:
  - a. Status of the project, complete/incomplete. If incomplete state what is incomplete. Failing to state the exact status or false or misleading statement will result a zero for the entire assignment.
  - b. Time spent on the project. - number of hours.
  - c. Problems encountered - List at least 3 to 4 biggest problems that you encountered while you were working on the project and how you solved them.
  - d. Things you wish you had been told prior to being given the assignment.
  - e. Observations- any interesting observation that you made about the runs described above.
2. Design Document (20%): The design document should be written prior to coding. There should **not** be any code in your design document. The goal of your design document should be for any programmer to be able to implement the project in any language using only your design document as a reference.
3. Code (10%): The portion of your code that relates to this part of the project. No extra piece of code should be submitted. You should also submit a Readme file explaining how to compile and run your tests.
4. Results (60%):
  - a. Report 1 and TRECEVAL report of each run.
  - b. Report 2 and TRECEVAL report of each run.