Midterm (30 pts)

1 Boolean Queries Complexity (8)

For a conjunctive query $kw_1 \land kw_2 \land \ldots \land kw_n$, is processing postings lists in the (increasing) order of size (length of the posting lists) guaranteed to be optimal? Explain why it is always the case, or give a counter-example where it is not.

2 True/False with Justification (2*3 + 2*2)

1. In a Boolean retrieval system, stop words elimination never lowers precision.

2. In a Boolean retrieval system, stemming never lowers recall.

3. Case-folding decreases the size of the dictionary.

4. Case-folding should be invoked at indexing time, but not while processing a query.

*Precision* is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search. *Recall* is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved).

3 Estimating Time for Sorting (4 + 2)

If we need $T \log_2 T$ comparisons (where $T$ is the number of termID-docID pairs) and two disk seeks for each comparison, how much time would index construction for the XYZ dataset take if we used disk instead of memory for storage and an unoptimized sorting algorithm (i.e., not an external sorting algorithm)? What is the closest time unit that captures the order of magnitude of the time required: a microsecond, a second, an hour, a day, a month or a year?
How does this compare to the case with the complete data in main memory and using in-memory sort (using the above scale)?

Use the system parameters and XYZ dataset parameters reproduced below. ($\log_{2}10 = 3.3$ and $\log_{10}2 = 0.3$)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>memory transfer time per byte</td>
<td>5 ns</td>
</tr>
<tr>
<td>s</td>
<td>average seek time</td>
<td>5 ms</td>
</tr>
<tr>
<td>b</td>
<td>disk transfer time per byte</td>
<td>20 ns</td>
</tr>
<tr>
<td>p</td>
<td>typical ALU operation</td>
<td>10 ns</td>
</tr>
<tr>
<td>M</td>
<td>Number of tokens</td>
<td>10 million</td>
</tr>
<tr>
<td>N</td>
<td>Number of terms</td>
<td>0.5 million</td>
</tr>
</tbody>
</table>

4 Citation Count Task (6)

Given a directed graph as an adjacency list: $\text{Ids} \rightarrow \text{list(Ids)}$

\[ s1 : s_{11}, s_{12}, ... \\
\text{s2} : s_{21}, s_{22}, ... \\
\text{...} \]

determine the list of pairs $(s_n, cnt)$, such that $cnt$ is the number of (in-)links incident on $s_n$, using MapReduce paradigm. E.g., for the following graph, the expected pairs are $(s_1, 1), (s_2, 1), (s_3, 1), (s_4, 2)$.

\[ s1 : s_2, s_4. \\
s2 : s_1, s_3, s_4. \]

Explain clearly the map task and the reduce task, defining these functions, preferably using set-notation.