Assignment 1: Index-List Compression (1 P.)

(a) Consider the $\gamma$ coding for the gaps in the documents ids in an inverted index list. Assume that the gaps are geometrically distributed, $P[\Delta = k] = (1 - p)^{k-1} \times p$ (strictly speaking, with truncation at some maximum possible gap, but you may disregard this aspect).

(i) First, decode the following sample list of $\gamma$-encoded gaps.

```
1110001110101011111101101111011
```

(ii) Then, compute the expected number of bits needed for encoding an index list with $n = 1001$ document ids. To do so, use the above assumption that gaps follow a geometric distribution, with parameter $p$, and the fact that a gap of size $k$ consumes $b(k) = 1 + 2\lceil\log_2 k\rceil$ bits. Fit parameter $p$ using the sample list of part (i).

(b) Download from the course website the file `data.txt.gz` and unzip it. It contains 100000 integer numbers. Implement Rice encoding and investigate the optimal choice of $M$ with respect to the compression ratio the encoding achieves compared to the raw representation of the numbers as unsigned int values (sizeof 4 bytes). Compare the determined optimal value of $M$ to the mean of the data and argue if the choice of $M$ is meaningful (or not). Further, if we know by inspecting the data that 175 is the maximum number in the data, how many bits are sufficient to represent a number and what compression ratio would be obtain compared to the original data—how does this compare to the optimal compression ratio computed by Rice before?

Assignment 2: Query Processing (1 P.)

(a) Consider a top-$k$ query with $m = 3$ terms, the user is interested in $k = 2$ results, and (non-weighted) summation as score aggregation. The underlying three index lists have the following (document identifier, score) entries:

<table>
<thead>
<tr>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$ 0.9</td>
<td>$d_3$ 0.8</td>
<td>$d_1$ 0.7</td>
</tr>
<tr>
<td>$d_4$ 0.5</td>
<td>$d_4$ 0.8</td>
<td>$d_6$ 0.6</td>
</tr>
<tr>
<td>$d_7$ 0.4</td>
<td>$d_7$ 0.7</td>
<td>$d_4$ 0.5</td>
</tr>
<tr>
<td>$d_2$ 0.3</td>
<td>$d_1$ 0.3</td>
<td>$d_7$ 0.5</td>
</tr>
<tr>
<td>$d_4$ 0.2</td>
<td>$d_6$ 0.2</td>
<td>$d_2$ 0.3</td>
</tr>
<tr>
<td>$d_5$ 0.2</td>
<td>$d_5$ 0.2</td>
<td>$d_3$ 0.1</td>
</tr>
<tr>
<td>$d_6$ 0.1</td>
<td>$d_2$ 0.2</td>
<td>$d_5$ 0.1</td>
</tr>
</tbody>
</table>

Apply the TA method (with random accesses) to this setting. Document all index accessing steps and the top-$k$ after each of them. How many sorted accesses (SA) and random accesses (RA) does the method need?

(b) Consider the following aggregation functions:

(i) $\max$

(ii) $\text{spread} = \max - \min$, where $\max$ and $\min$ refer to the scores of the same document from different lists.
Assignment 3: Aggr. Functions

An $m$-ary aggregation function $f : [0,1]^m \rightarrow [0,1]$ is said to be strict if $f(x_1, \ldots, x_m) = 1 \Leftrightarrow x_1 = 1 \land \ldots \land x_m = 1$. $f$ is monotone if $(x_1 \leq x'_1 \land \ldots \land x_m \leq x'_m) \Rightarrow (f(x_1, \ldots, x_m) \leq f(x'_1, \ldots, x'_m))$. A binary aggregation function $f : [0,1] \times [0,1] \rightarrow [0,1]$ is called a triangular norm if the following four properties hold:

1. $f(0, 0) = 0 \land f(x, 1) = f(1, x) = x$
2. $(x_1 \leq x'_1 \land x_2 \leq x'_2) \Rightarrow (f(x_1, x_2) \leq f(x'_1, x'_2))$
3. $f(x_1, x_2) = f(x_2, x_1)$
4. $f(f(x_1, x_2), x_3) = f(x_1, f(x_2, x_3))$

Which of the following aggregation functions are monotone, which ones are strict, and which correspond to the triangular norm?

(i) $\min$

(ii) bounded sum: $f(x_1, x_2) = \min(1, x_1 + x_2)$