Solutions to Problem Set 1

Problem 1. The original integers and their respective γ encodings are:

(i) $1 \rightarrow 0$

(ii) $38 \rightarrow 11111000110$

(iii) $1026 \rightarrow 111111111100000000010$

Problem 2. This questions asks you to compare three different encodings for postings lists. For a given positive integer N, unary encoding uses N bits, fixed-length binary encoding for a $50,000$-document collection uses $\lceil \log_2 50,000 \rceil$ = 16 bits, and gamma-encoding uses $2\lfloor \log_2 N \rfloor + 1$ bits. Using these expressions, we compute the overall size as follows:

- *Case 1.* In this case, the size of the postings list is the sum of the sizes of the encodings of 3, 5, 1524, 1600, 29307, and 31520.
  - Unary: $3 + 5 + 1524 + 1600 + 29307 + 31520$ = 63959 bits $\approx$ 8000 bytes
  - Fixed-length binary: $6 \star 16$ = 96 bits = 12 bytes
  - Gamma-encoded: $3 + 5 + 21 + 21 + 29 + 29$ = 108 bits $\approx$ 14 bytes
- *Case 2.* In this case, the size of the postings list is the sum of the sizes of the encodings of the gaps, i.e., 3, 2, 1519, 76, 27707, and 2213.
  - Unary: $3 + 2 + 1519 + 76 + 27707 + 2213$ = 31520 bits = 3940 bytes
  - Fixed-length binary: $6 \star 16$ = 96 bits = 12 bytes
  - Gamma-encoded: $3 + 3 + 21 + 13 + 29 + 23$ = 92 bits $\approx$ 12 bytes

Problem 3. As in the lecture notes, we will assume that the most frequent term has a postings list with n gaps of 1 each, the second most frequent with $\frac{n}{2}$ gaps of 2 each, etc. Here, n = 2 million and the number of distinct terms m = 500K.

(i) The postings list for the k^{th} most frequent term takes $\frac{n}{k} \times k = n$ bits. Hence total size is $m \times n \approx$ 100 GB.

(ii) The postings list for the k^{th} most frequent term takes $\frac{n}{k}(2\lfloor \log_2 k \rfloor + 1)$ bits. Hence, we need to compute

$$
\begin{aligned}
\sum_{k=1}^{500K} \frac{n}{k}(2\lfloor \log_2 k \rfloor + 1) &\approx \sum_{i=1}^{19} \sum_{k=2^{i-1}}^{k<2^i} \frac{n}{k}(2\lfloor \log_2 k \rfloor + 1) \\
&= \sum_{i=1}^{19} n[2(i-1)+1] \sum_{k=2^{i-1}}^{k<2^i} \frac{1}{k} \\
&< \sum_{i=1}^{19} n[2(i-1)+1] \left\{ \frac{2^i - 2^{i-1}}{2^{i-1}} \right\} \\
&= 2n \sum_{i=1}^{19} i - \sum_{i=1}^{19} n \\
&= 361\, n\ Mbits \\
&\approx 90MB
\end{aligned}
$$

(iii) We now need to compute $\sum_{k=101}^{500K} \frac{n}{k}(2\lfloor \log_2 k \rfloor + 1)$. Since 101 falls between 2^6 and 2^7, we will estimate this summation using the same technique as in (ii) but using both $i = 6 \ldots 19$ and $i = 7 \ldots 19$. Using $i = 6 \ldots 19$, we get $2n \sum_{i=6}^{19} i - \sum_{i=6}^{19} n =$ $336\, n\ Mbits \approx$ 84 MB. Using $i = 7 \ldots 19$, we get $325\, n\ Mbits \approx$ 81 MB. Hence, we estimate the size of the postings file to be between 81 and 84 MB.

Problem 4. We used the stemmer at `http://maya.cs.depaul.edu/~mobasher/classes/ds575/porter.html`.

Term	Stem
automobile	automobil
automotive	automot
cars	car
information	inform
informative	inform

Problem 5. Recall the query processing heuristic of choosing terms in increasing order of their frequency when computing an AND query. The frequency of an OR expression is estimated to be the sum of the frequency of its individual components. In addition, for a term t, $NOT\, t$ has an associated frequency of $N - frequency(t)$, where N is the total number of documents. Using these rules:

(i) ((kaleidoscope OR eyes) AND (tangerine OR trees)) AND (marmalade or skies)

(ii) (tangerine AND (NOT trees)) AND (NOT marmalade)

Problem 6. The occurrences of *The undiscovered country* are listed in the following table.

Document	Occurrences
5	2 at (16,17,18) and (44,45,46)
7	1 at (67,68,69)

Problem 7. Let's first determine how much storage is needed for k terms, using a block size of k (for arbitrary k). Then we can simply plug in $k = \{4, 8, 16\}$.

Based on Lecture 2, slide 9, we develop the following for how much space is required for k terms:

- $4 \times k$ bytes for Freq
- $4 \times k$ bytes for Postings ptr
- 3 bytes for term pointer (note, this is NOT $3 \times k$)
- $(8 + 1) \times k$ bytes for k terms in term string (the additional 1 byte is for the term length which becomes necessary under blocking)

So k terms require $4k + 4k + 9k + 3 = 17k + 3$ bytes. So the amortized per term space requirement is $17 + \frac{3}{k}$ bytes. The total space required for the dictionary is then given by $500,000$ terms $\times(17 + \frac{3}{k})$:

k	Space
4	8.88 MB
8	8.69 MB
16	8.59 MB

Larger values of k yield little benefit, as even with the largest block size possible, $500,000 * 17 = 8.5$ MB will be required.