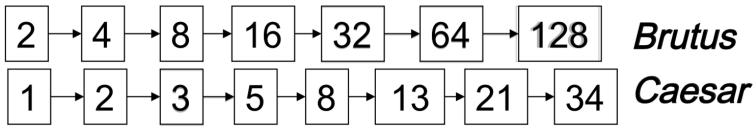
Assignment #1

B05902120 / Yu-Ting, TSENG

Mar 19, 2019

Query Processing

Consider the Boolean query X AND NOT Y. Assume x and y are the lengths of the postings lists for X and Y, respectively, and N is the number of documents in total. A naive evaluation of the query X AND NOT Y would be to calculate (NOT Y) first as a new postings list, which takes $O(N)$ time, and the merge it with the postings list for X. Therefore, the overall complexity will be $O(N)$. Please write out a postings merge algorithm that evaluates this query in time $O(x + y)$.



Supposed the first line is the sorted posting list of X, and the pointer pointing to the element in list X is `p_x`; the second is the one of Y, and the pointer is `p_y`. We go through the list X and Y in order. There are some rules to obey. 1. When the value of `p_y` is bigger than the one of `p_x`, move `p_x` and add the elements which have been go through in this step into result list till the situation vanish. 2. If two values are equal, move two pointers to the next elements. 3. In the last case, move `p_y` to the next element. Repeatedly execute those three steps until the list have been completely gone through. As a result the time complexity would be $O(x + y)$.

Zipf's Law

Assume that the frequency distribution of words in a collection of documents C roughly follows the Zipf's law: $r * (w_r|C) = 0.1$, where $r = 1, 2, 3, ...$ is the rank of a word in the descending order of frequency. w_r is the word at rank r, and $P(w_r|C)$ is the probability (frequency) of word w_r in the collection. What is the probability of the most frequent word in the collection? What is the probability of the second most frequent word in the

collection? (When calculating the probabilities for more words, you will find that the top 50 most frequent words account for about 45%. That's nearly a half of the text!)

The probability of the most frequent word in the collection would be $0.1/1 = 0.1$. The probability of the second most frequent word in the collection would be $0.1/2 = 0.05$.

Zipf's Law

From the following sequence of γ-coded gaps, reconstruct the gap sequence and then the postings sequence:

$$1110001110101011111101101111011.$$

To begin, we can separate the sequence into several parts: 1110001, 11010, 101, 11111011011, 11011. According to Zipf's Law, we can decode them as 9, 6, 3, 59, 7, which represent the gaps. As a result, the answer will be 9, 15, 18, 77, 84.

Skip Pointer

Just as what we discussed in class, the optimal skip distance c can be determined by minimizing the quantity $kn/c + pc/2$, where k is the skip pointer length, n is the total inverted list size, c is the skip interval, and p is the number of postings to find. Plot this function using $k = 4$, $n = 1000000$ and $p = 1000$, but varying c. Then, plot the same function, but set $p = 10000$. Describe how the optimal value for c changes. Finally, take the derivative of the function $kn/c + pc/2$ in terms of c to find the optimum value for c for a given set of other parameters, i.e., k, n, and p.

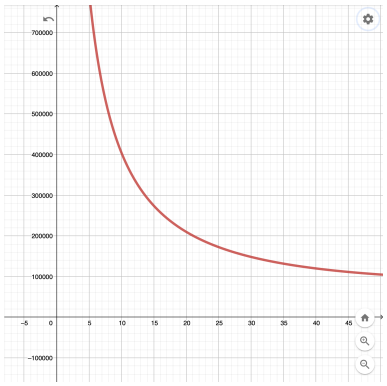


Figure 1: $p = 1000$

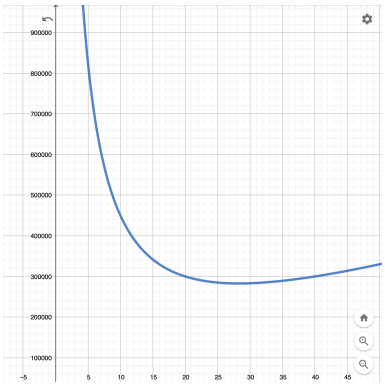


Figure 2: $p = 10000$

Derivative result:

$$\frac{\partial}{\partial c}\{kn/c + pc/2\} = p/2 - kn/c^2$$

According to the second derivative, when $c > 0$, the result value is always positive. The optimize value happens when $c = \sqrt{2kn/p}$.