Web Retrieval and Mining Assignment #1

B0591025 Yu-Neng Wang

March 27, 2020

Problem

1. **Query Processing:**

   We maintain two pointers, p_x and p_y, into X and Y and walk through Y and X. At each step, we advance p_y and check whether $docID(p_x) < docID(p_y)$. If so, put $docID(p_x)$ into resulting list and advance p_x. If $docID(p_x)$ equals $docID(p_y)$, advance p_x. Otherwise, stop p_x and continue next step of advance p_y. Looping the above step until run over X and Y (or if p_y is NIL while p_x is not, we put remaining $docIDs$ in X into resulting list). Because we only visit every element in X and Y once, this merging algorithm evaluate the query in time $O(x + y)$ where x and y are length of two lists..

2. **Zip's Law:**

   - The probability of the most frequent word in the collection: $p(w_1|C) = 0.1/1 = 0.1$
   - The probability of the second-most frequent word in the collection: $p(w_2|C) = 0.1/2 = 0.05$

3. γ **code: 1110001110101011111101101111011**

   This γ code can be decoded as follows:

   $$\Rightarrow 1110001, 11010, 101, 11111011011, 11011$$
   $$\Rightarrow 9, 6, 3, 59, 7 \ (gap\ sequence)$$
   $$\Rightarrow 9, 15, 18, 77, 84 \ (posting\ sequence)$$

4. **Skip Pointer:**

   As p increases by 10x, the optimal value for c becomes $\frac{1}{\sqrt{10}}$, which is inversely proportional to $\sqrt{p}$. That is to say, to find more posting with minimal bytes read, others the same, one should reduce skip distance c.

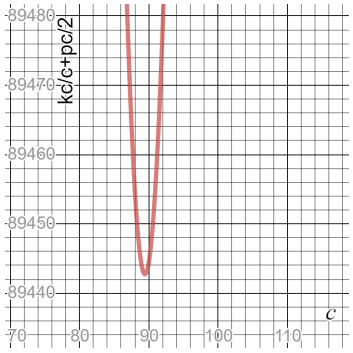

Figure 1: k=4, n=1000000, p=1000

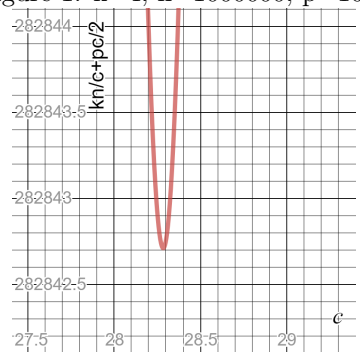

Figure 2: k=4, n=1000000, p=10000

The optimal value of c for a given set of k, n, and p can be derived as follow:

$$\begin{aligned} & \frac{d}{dc}\left(\frac{kn}{c}+\frac{pc}{2}\right)=0 \\ \Rightarrow\ & -\frac{kn}{c^2}+\frac{p}{2}=0 \\ \Rightarrow\ & c^2=\frac{2kn}{p} \\ \Rightarrow\ & c=\sqrt{\frac{2kn}{p}}\ (c>0) \end{aligned}$$