1 Finger Printing

1.1 Problem: Database Consistency

Given two databases $A, B$ check if they are consistent. The resource of interest is the communication complexity.

Assume that

$A = a_{n-1}, a_{n-2}, \ldots, a_0$

$B = b_{n-1}, b_{n-2}, \ldots, b_0$

the two databases in bits. We can solve this problem with a communication complexity of $O(n)$.

Randomized Algorithm:

Idea: Let

$A' = \sum_{i=0}^{n-1} a_i 2^i$

and

$B' = \sum_{i=0}^{n-1} b_i 2^i$

Algorithm:

- Pick a random prime $p$
- Check if $A' \mod p = B' \mod p$

What is the probability $\text{Prob}[A' \mod p = B' \mod p | A' \neq B']$?

Let $C = |A' - B'|$ then

$A' \mod p = B' \mod p \Rightarrow C \mod p = 0 \Rightarrow p | C$

The number of prime factors for $C$ is $\leq n$ since each prime number is $\geq 2$. Let $p$ be picked from the interval $[2, \tau]$. 

1
Theorem 1.1 Prime number Theorem: The number of primes in $[2, \tau]$ is $\Theta \left( \frac{\tau}{\log \tau} \right)$.

From the theorem follows that:

$$\text{Prob}[A' \mod p = B' \mod p | A' \neq B'] = O \left( \frac{n}{\tau \log \tau} \right)$$

If $\tau = nt \log nt$ then the above probability is equal to $O(\frac{1}{t})$.

1.2 Problem: String Matching

Input:
A text

\[ T = t_1 t_2 \ldots t_n \]

and a pattern

\[ P = p_1 p_2 \ldots p_m \]

with $m \ll n$.

Output:
All $j$’s such that $t_j t_{j+1} \ldots t_{j+m-1} = P$. Let

\[ T_j = t_j t_{j+1} \ldots t_{j+m-1} \]

Assume w.l.o.g. that the alphabet of concern is $\Sigma = \{0, 1\}$.

1: \textbf{for} i=0 to (n-m+1) \textbf{do}
2: \hspace{1em} Use the algorithm in Section 1.1 to check if $T_j = P$.
3: \textbf{end for}

For a specific $j$,

$$\text{Prob}[T_j \mod p = P \mod p | T_j \neq P] = O \left( \frac{m}{\tau \log \tau} \right) \tag{1}$$

$$\Rightarrow \text{Prob}[\exists j : T_j \mod p = P \mod p | T_j \neq P] = O \left( \frac{nm}{\tau \log \tau} \right) \tag{2}$$

If we pick $\tau = nmt \log(nmt)$ then the above probability is $O(\frac{1}{t})$.

We can obtain a Las Vegas algorithm from the above by checking for a match rigorously whenever $T_j \mod p = P \mod p$. The worst case run time is $O(mn)$.
\[
T_j = t_j t_{j+1} \ldots t_{j+m-1}
\]
\[
T_{j+1} = t_{j+1} t_{j+2} \ldots t_{j+m}
\]
\[
T_{j} = 2^{m-1} t_j + 2^{m-2} t_{j+1} + \ldots + 2 t_{j+m-2} + t_{j+m-1}
\]
\[
T_{j+1} = 2^{m-1} t_{j+1} + 2^{m-2} t_{j+2} + \ldots + 2 t_{j+m-1} + t_{j+m}
\]
\[
\Rightarrow T_{j+1} = 2 T_j - 2^m t_j + t_{j+m}
\]

Computing each \( T_j \) takes \( O(1) \) time. Run time of the algorithm is \( O(n + m) \).

2 A Randomized Data Structure

**Goal:** Implement a dictionary. Use a skip list

Let

\[
S = k_1, k_2, \ldots, k_n
\]

be the input set.

**Definition 2.1** A leveling of \( S \) with \( r \) levels will be a sequence

\[
L_1 \subseteq L_2 \subseteq \ldots \subseteq L_r
\]

with \( L_1 = S \) and \( L_r = \emptyset \).

**Definition 2.2** The level of any element \( x = \ell(x) \) is

\[
\ell(x) = \{ \max i : x \in L_i \}
\]

Assume that \( -\infty, \infty \) are in each \( L_i \).

An interval in level \( j \) is the set of elements spanned by two successive elements in this level. We can construct a tree using the intervals (see Figure 1).

For any element \( y \), let \( I_j(y) \) be the interval in level \( j \) that contains \( y \). Consider

\[
I_r(y), I_{r-1}(y), \ldots, I_1(y)
\]

This is a root to leaf path.
Figure 1: Tree Construction.

**Run Time for Searching $y$:**

$$\sum_{j=1}^{r} [1 + c(I_j(y))]$$

where $c(I_j(y))$ is the number of children of the interval $I_j(y)$. To construct the level $L_{i+1}$ from $L_i$ we follow the steps below:

- for each element $x \in L_i$, include $x$ in $L_{i+1}$ with prob. 1/2.

For a given $x$,

$$\text{Prob}[\ell(x) > q] \leq 2^{-q} \Rightarrow \text{Prob}[\exists x : \ell(x) > q] \leq n2^{-q}$$

We want: $n2^{-q} \leq n^{-\alpha}$. Equating the two:

$$q = (\alpha + 1) \log n$$

So the length of the interval tree is $\tilde{O}(\log n)$.

Let $I$ be any interval in level $j$. Let $[x_1, x_2], [x_3, x_3], \ldots, [x_s, x_{s+1}]$ be the right siblings of $I$ in level $j$. Note that the elements $x_1, x_2, \ldots, x_{s+1}$ belong to level $j$, but they are not in level $j + 1$. This happens with probability $2^{-s}$ and so we have a geometric distribution with parameter 1/2. Hence the expected value of $s$ is equal to 2.