# CSE 6512 Lecture 7 Notes
Sudipta Pathak
September 20, 2011

## Pattern Matching:

INPUT : $T = t_1t_2t_3......................t_n \in \Sigma^*$
$P = p_1p_2p_3..................p_m \in \Sigma^*$

OUTPUT: All the indices i such that $T_i = t_it_{i+1}t_{i+2}........t_{i+m-1} = P$

Algorithm : for i= 1 to (n-m+1) do
check if $T_i = P$
using the previous algorithm (for checking the equality of two integers)
if yes, output i;

Analysis : Let the prime be picked from the interval [1,k] =>
\# of such primes = $\Theta$ (k/log k)

probability of an incorrect answer for a specific i= m/(k /log k)
=> probability of an incorrect answer for at least one such i is $\leq$ n.m/(k /log k)

we want this to be $\leq n^{-\alpha}$

$$\Rightarrow n.m/(k / \log k) = n^{-\alpha}$$

$$\Rightarrow m.n^{\alpha+1} = k/(\log k)$$

pick k to be $( m.n^{\alpha+1}) \log ( m.n^{\alpha+1}) = \Omega (m.n^{\alpha+1} \log n)$.

Note : $T_i = 2^{m-1}t_i + 2^{m-2}t_{i+1} + \ldots\ldots\ldots\ldots+ 2t_{i+m-2} + t_{i+m-1}$
$T_{i+1} = 2^{m-1}t_{i+1} + 2^{m-2}t_{i+2} + \ldots\ldots\ldots\ldots+ 2t_{i+m-1} + t_{i+m}$
$2T_i = 2^m t_i + 2^{m-1}t_{i+1} + 2^{m-2}t_{i+2} + \ldots\ldots\ldots+ 2t_{i+m-1}$

$T_{i+1} = 2T_i - 2^m t_i + t_{i+m}$.

The above equality implies that for each i ($1\leq i\leq n-m+1$), checking if $T_i = P$ takes only O(1) time.
As a result, the total runtime is = O(n).

We can convert this into a Las Vegas algorithm by brute force checking for every

“HIT”. A “HIT” occurs for position i if $T_i$ mod p = P mod p, where p is the prime number used. The worst case runtime of this algorithm is $\Omega$ (m.n).

**RANDOMIZED SKIP LIST:**

A randomized skip list is a data structure that can be used to realize a dictionary, i.e., a data structure that supports these three operations: SEARCH, INSERT, and DELETE.

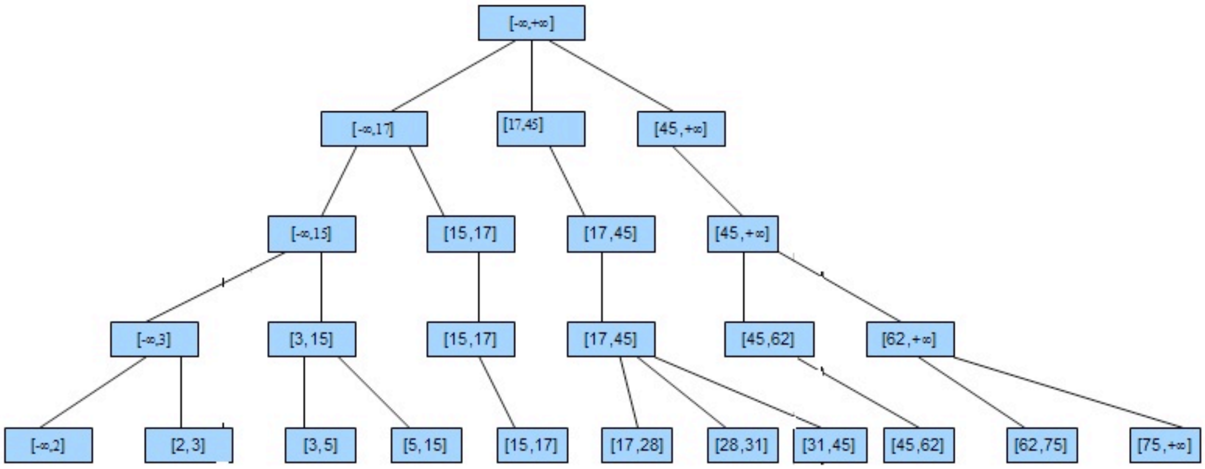
Let S be a given ordered set.
A leveling of S with r levels is a sequence :
$L_r \subseteq L_{r-1} \subseteq \cdots \subseteq L_2 \subseteq L_1$ where $L_1 = S$ & $L_r = \Phi$

Definition:
The level of any element x is $\ell(x)$ = Max i such that x $\in L_i$.

Definition :
An interval at any level is nothing but an interval of two successive elements. The following is an example where S = {2, 3, 5, 15, 17, 28, 31, 45, 62, 75}. Assume that the two elements -∞ and +∞ are members of each level. Using the intervals of the different levels we can construct a tree as shown below.

```
-∞------------------------------------------------------------------------------------------ +∞

-∞----------------------------------- 17 ------------------------ 45 ------------------------ +∞

-∞-------------------------- 15 ----- 17 ------------------------ 45 ------------------------ +∞

-∞------------ 3----------- 15 ----- 17 ------------------------ 45 ------ 62 ------------- +∞

-∞----- 2 ----- 3----- 5----- 15 ----- 17 ----- 28 ------31 ------ 45 ------ 62 ------ 75 --- +∞
```

TREE :




Definition :

For any element x, let $I_j(x)$ stand for the interval that x belongs in level j.

SEARCH(x):

Go through : $I_r(x), I_{r-1}(x), I_{r-2}(x), \ldots\ldots\ldots\ldots$, until the answer is found.

TIME NEEDED : $\sum_{j=r}^{1} c(I_j(x))$ where $c(I_j(x))$ is the # of children of $I_j(x)$.

$\text{Prob}[\text{level}(x) = h] = (\tfrac{1}{2})^{h-1}(1/2) = (1/2)^h$

$\text{Prob}[\text{level}(x) > h] = (\tfrac{1}{2})^{h+1}[1 + \tfrac{1}{2} + 1/4 + \ldots\ldots] \leq (\tfrac{1}{2})^h$

$\text{Prob}[\exists\ x \text{ whose height is} > h] \leq n\,(\tfrac{1}{2})^h$

we want this to be $\leq n^{-\alpha}$

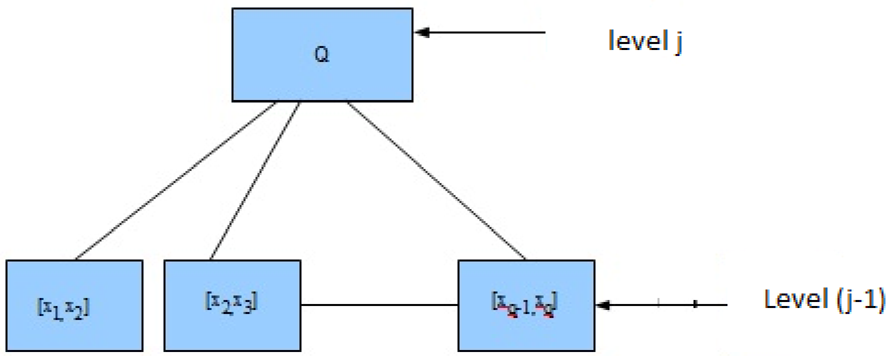
$\Rightarrow n^{-\alpha} = n\,(\tfrac{1}{2})^h$

$\Rightarrow 2^h = n^{\alpha+1}$

$\Rightarrow h = (\alpha+1)\log(n)$

=> The height of the tree is Õ(log n)

What is $E\left[\sum_{j=r}^{1} c(I_j(x))\right]$ ?




If some node Q at level j has q children, this could only be because the elements $x_2,\ldots\ldots,x_{q-1}$ were not picked to be in $L_j$ & they were in $L_{j-1}$. The # of such elements (that are not in $L_j$) is upper bounded by a Geometric Distribution with parameter ½.
=> the expected value = 2

=> $E\left[c_j(I)\right] = O(1)$ for any interval I

=> $E\left[\sum_{j=r}^{1} c(I_j(x))\right]$

=> $(1- n^{-\alpha})O(\log n)O(1) + n^{-\alpha}.O(n)$
=> $O(\log n)$

$$\boxed{E[A] = E[A/B]\Pr[B] + E[A/\overline{B}]\Pr[\overline{B}]}$$

<u>INSERT(x):</u>
Pick a random level for x. If $\ell(x) > r$ increment r by 1. Use the search algorithm to find a relevant place for x. Some of the intervals may have to be split.

Expected time = O(log n).
Delete also is processed likewise.

Theorem : In a random skiplist we can perform the following operations in an expected O(log n) time : SEARCH, INSERT, and DELETE.