# Class Lecture – Eight (8)

# Randomized Hashing

Two data structures of interest are:

1) Static Dictionary: where a set of keys S is given and we have to organize it into a data structure that supports the efficient processing of FIND operations.
2) In a Dynamic Dictionary the set is not provided in advance rather it is constructed by a series of INSERT and DELETE operations that are intermingled with the FIND operations.

The above can be realized using balanced search trees, random treaps, random skip lists, etc. But in the worst-case, for a set S of size s, they require $\Omega(\log s)$ time to process any search or update operation. When the keys are general, this is a lower bound on the search time.

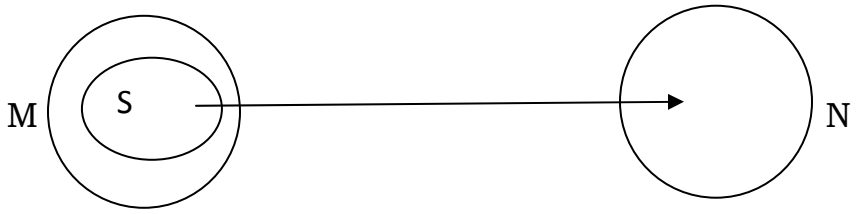
We can do better if additional information is known about the keys. One such case is when the keys in S are integers from a universe M where M = {0, 1, 2, …, m-1}. If we have O(m) memory, we can use a bit array of size m and perform the dictionary operations in O(1) time. If m is very large and we have only have O(n) space, we can use hashing. The idea is to use an array T[0:n-1] of lists and a hash function h:M->N where N={0,1, … ,n-1}. If x is any element in the data structure it will be stored in the list T[h(x)]. If the elements of S are assumed to be randomly picked from M and if n=cs for some constant c, then the expected length of each of the lists in T will be O(1) and hence each of the dictionary operations can be performed in an expected O(1) time. This expectation is in the space of all possible inputs.

We can achieve a similar performance using randomization. In this case we will not assume uniformity in the input space.

**Definition 1**: A collision is said to occur between two distinct keys x and y if h(x) = h(y) and they are said to collide at the corresponding location in T.

**Definition 2**: A hash function h : M $\rightarrow$ N is said to be perfect for a set S $\subseteq$ M if h does not cause any collisions among the keys of the set S.

**Fact:** For any given set S $\subseteq$ M we can always find a hash function h that is perfect for S.




Proof: Let |S| = s $\leq$ |N|. So, the total number of hash functions from M to N = $n^m$. The number of hash functions that are perfect for S = $\binom{n}{s}$s!. □

**Fact:** For any hash function h, we can find a S $\subseteq$ M such that h is not perfect for S.

Proof: Since N>M, at least two elements x and y of M will collide under any h. Include these two in S. □

Since a single hash function may not be perfect for each possible S, we can use a family of hash functions. Also, we may insist on near perfectness rather than perfectness. If for each possible S, a "good" number of hash functions in the family is near perfect, then if we pick a random member of the hash family, it will be near perfect for the given S with some "good" probability.

**Definition 3:** Let M = {0, 1, 2, …, M - 1} and N = {0,1, 2, …, n - 1}, with m $\geq$ n. A family H of functions from M into N is said to be 2-universal if, for all x, y $\subseteq$ M such that x $\neq$ y, and for h chosen uniformly at random from H, Prob [h(x) = h(y)] $\leq \frac{1}{n}$.

**Note:** The set of all functions from M to N has this property. A totally random mapping from M to N has a collision probability of exactly 1/n;

thus, a random choice from a 2-univarsal family of hash functions gives a seemingly random function. But there are $n^m$ functions in that family. This calls for $\Omega(m * \log n)$ memory (even to specify a member of the family).

**Definition 4:** For any $x, y \subseteq M$ and $h \subseteq H$, define the following indicator function for a collision between the keys x, y under the hash function h:

$$\delta(x, y, h) = \begin{cases} 1 \text{ for } h(x) = h(y) \text{ and } x \neq y \\ 0 \text{ otherwise} \end{cases}$$

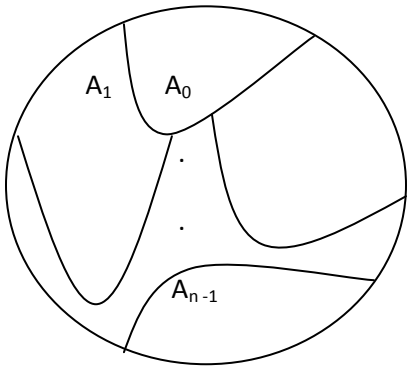
For all $X, Y \subseteq M$, the following extensions of the indicator function $\delta$ can be defined as:

$$\delta(x, y, H) = \sum_{h \in H} \delta(x, y, h) \ ,$$
$$\delta(x, Y, h) = \sum_{y \in Y} \delta(x, y, h) \ ,$$
$$\delta(X, Y, h) = \sum_{x \in X} \delta(x, Y, h) \ ,$$
$$\delta(x, Y, H) = \sum_{y \in Y} \delta(x, y, H) \ ,$$
$$\delta(X, Y, H) = \sum_{h \in H} \delta(X, Y, h) \ .$$

For a 2-universal family H and $x \neq y$, we have $\delta(x, y, H) \leq |H|/n$.

**Lemma 1:** Let, H be any family of hash functions, then we can always find $x, y \subseteq M$ such that $\delta(x, y, H) \geq \frac{|H|}{n} - \frac{|H|}{m}$.

**Proof:**
Let h be any member of H and $A_z = \{ x \in M: h(x) = z \}$ for $z = \{0, 1, .., n-1\}$. The sets $A_z$, for $z \in N$, form a partition of M.




It is easy to see that,

$$\delta(M, M, h) = \sum_{z=0}^{n-1} |Az|(|Az|-1).$$

This sum is minimal if all the $A_{z'}$s are of the same size, i.e., m/n. So, we obtain:

$$\delta(M, M, h) \geq n(\frac{m}{n}(\frac{m}{n} - 1)) = m^2(\frac{1}{n} - \frac{1}{m})$$

By the pigeonhole principle there must exists a pair of elements x, y $\subseteq$ M such that: $\delta(x, y, H) \geq \frac{\delta(M, M, H)}{m^2} = \frac{|H|\delta(M, M, h)}{m^2} \geq \frac{|H|m^2(1/n - 1/m)}{m^2}$

$$= |H|(\frac{1}{n} - \frac{1}{m}). \square$$

**Lemma 2:** Let, S be any subset of M and H be a 2-universal family of hash functions. If h is randomly picked from H, then $E[\delta(x, S, h)] \leq |S|/n$, for any x from M.

Proof:

$$E[\delta(x, S, h)] = \sum_{h \in H} \frac{\delta(x, S, h)}{|H|} = \frac{1}{|H|} \sum_{h \in H} \sum_{y \in S} \delta(x, y, h)$$

$$= \frac{1}{|H|} \sum_{y \in S} \sum_{h \in H} \delta(x, y, h)$$

$$= \frac{1}{|H|} \sum_{y \in S} \delta(x, y, H)$$

$$\leq \frac{1}{|H|} \sum_{y \in S} \frac{|H|}{n}$$

$$= \frac{|S|}{n} . \square$$

**Lemma 3:** If there are q dictionary operations, then the expected time to process them will be $O(q(1+\frac{s}{n}))$ where s is the max number of INSERT operations. □

**Construction of a 2-universal family of hash functions:**

Pick prime number $p \geq m$. Use the field $Z_p = \{0, 1, .., p-1\}$.
Let, $f_{a,b}(x) = (ax + b) \bmod p$ for $a, b \in Z_p$ with $a \neq 0$ and $g(x) = x \bmod n$; define $h_{a,b}(x) = g(h_{a,b}(x)) = (ax + b) \bmod p \bmod n$.
Let $H = \{h_{a,b}(x) \mid a, b \in Z_p, a \neq 0\}$
Consequently, $|H| = p(p-1)$. Note that any member of H can be specified with O(log m) bits.
**Lemma 4:** The above H is 2-universal.
Proof: in the next lecture.

**Prepared By:** *Subrata Saha*