1 Hash Table

The purpose of hash table is maintaining a (possibly evolving) set of stuff, for example transactions, people + associated data, IP address etc. In all these examples, there is a universe $U$ of possible elements that is extremely large and the hash table is trying to keep track of a set $S \subseteq U$ whose size is generally a negligible fraction of $U$. And the goal is to do following operations:

- Insert $(S, x) : S \leftarrow S \cup \{x\}$
- Delete $(S, x) : S \leftarrow S - \{x\}$
- Search $(S, k) : \text{return } x \text{ such that } key[x] = k \text{ or return } \text{nil} \text{ if no such } x.$

2 Hashing Function

**Definition 1.** Hash function $h : U \rightarrow \{0, 1, \ldots, n - 1\}$ maps keys “randomly” onto a table.

When a record to be inserted maps to an already occupied space, a collision occurs. To resolve collision, the idea is to link record in same slot in a list.

**Definition 2.** The load factor of a hash table with $n$ keys and $m$ slots is $\alpha = \frac{n}{m} = \text{ave } \# \text{ keys per slot.}$
• worst case: given $|S| = n$, every key hashes to the same slot and access takes $O(n)$ time

• average case: expected unsuccessful searching time is $O(1 + \alpha)$

In sum, the properties of a good hash function can be summarized as:

• should distribute keys uniformly into slots
• should be easy to store and very fast to evaluate

3 A 2-Universal Hashing Example

To hash 9-digit student ID, we divide the original number into three parts $d_1, d_2, d_3$.

• pick prime $n$ (little bit larger than number of students)
  - $n$ should not be too close to power of 2
  - $n$ should not be too close to power of 10

• randomly pick $a_1, a_2, a_3 \in \{0, 1, \ldots, n - 1\}$ and define
  \[
  h(d_1, d_2, d_3) = a_1d_1 + a_2d_2 + a_3d_3 \pmod{n}
  \]

Theorem 3. For any keys $x$ and $y$, this function will evenly distribute keys such that
\[
P(h(x) = h(y)) = \frac{1}{n}.
\]

Proof: Suppose $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$ are distinct, without loss of generalization, assume $x_3 \neq y_3$. Given $h(x_1, x_2, x_3) = h(y_1, y_2, y_3)$, we have

\[
a_1x_1 + a_2x_2 + a_3x_3 = a_1y_1 + a_2y_2 + a_3y_3 \pmod{n}
\]
\[
\Rightarrow a_1(x_1 - y_1) + a_2(x_2 - y_2) = a_3(y_3 - x_3) \pmod{n}
\]
\[
\Rightarrow a_3 = a_1\frac{(x_1 - y_1)}{y_3 - x_3} + a_2\frac{(x_2 - y_2)}{y_3 - x_3} \pmod{n}
\]

The chance of picking the right $a_3$ is $\frac{1}{n}$. \square

For perfectly randomized function $h(x)$

\[
P(h(x_1) = h(x_2)) = \sum_{k=0}^{n-1} P(h(x_1) = h(x_2) = k) = \sum_{k=0}^{n-1} \frac{1}{n^2} = \frac{1}{n}.
\]