1.3.3 Universal hashing

If a malicious adversary chooses the keys to be hashed, then he can choose n keys that all hash to the same slot, yielding an average retrieval time of O(n). Any fixed hash function is vulnerable to this sort of worst-case behavior; the only effective way to improve the situation is to choose the hash function randomly in a way that is independent of the keys that are actually going to be stored. This approach, called universal hashing, yields good performance on the average, no matter what keys are chosen by the adversary.

Definition A randomized algorithm H for constructing hash functions h : U → {1,... ,M} is universal if for all $x \neq y$ in U, we have

$$\Pr_{h \leftarrow H}[h(x) = h(y)] \leq 1/M.$$

We also say that a set H of hash functions is a universal hash function family if the procedure “choose h ∈ H at random” is universal. (Here we are identifying the set of functions with the uniform distribution over the set.)

The following theorem shows that a universal class of hash functions gives good average-case behavior.

Theorem 1.3

If H is universal, then for any set $S \subseteq U$ of size N, for any $x \in U$ (e.g., that we might want to lookup), if we construct h at random according to H, the expected number of collisions between x and other elements in S is at most N/M.

Proof: Each $y \in S$ $(y \neq x)$ has at most a 1/M chance of colliding with x by the definition of “universal”. So,

- Let C_{xy} = 1 if x and y collide and 0 otherwise.
- Let C_x denote the total number of collisions for x. So, $C_x = \sum_{y \in S, y \neq x} C_{xy}$.
- We know E[C_{xy}] = Pr(x and y collide) ≤ 1/M.
- So, by linearity of expectation, E[C_x] = $\sum_y E[C_{xy}]$ < N/M.

Corollary

If H is universal then for any sequence of L insert, lookup, and delete operations in which there are at most M elements in the system at any one time, the expected total cost of the L operations for a random h $\in$ H is only O(L) (viewing the time to compute h as constant)

Proof : For any given operation in the sequence, its expected cost is constant by Theorem 1.2, so the expected total cost of the L operations is O(L) by linearity of expectation.

See Theorem 11.5 form Book -T. H. Cormen, C. E. Leiserson , R. L. Rivest and C. Stein : Introduction to Algorithms, Third Edition ,The MIT Press Cambridge, Massachusetts London, England .

Reference :

Text Book-T. H. Cormen, C. E. Leiserson , R. L. Rivest and C. Stein : Introduction to Algorithms, Third Edition ,The MIT Press Cambridge, Massachusetts London, England .