Date: Aug 7, 2025

Hash table

- It is a data structure in which keys are mapped to array positions by a hash function.
- Let U be the universe of keys and K be the keys that are actually in use. The storage requirement for a hash table is $O(K)$.
- In a hash table, an element E_k is stored at index $H(k)$.

Hashing

- It is the process of mapping the keys to appropriate locations or indices in a hash table.

Collision

- When more than one keys generate the same index then collision is said to occur.

Hash function

- It is a mathematical formula, which when applied to E, produces an integer which can be used as an index for the key in the hash table.

Properties of a good hash function

- Low cost: The cost of executing a hash function must be small.
- Deterministic: A hash procedure must be deterministic, i.e. the same hash value must be generated for a given input value.
- Uniformity: A good hash function must map the keys as evenly as possible over its output range so that the number of collision may be minimized.

Different hash functions

1. Division method

   $h(x) = x \mod M$

   where x is the key and M is the size of any number (preferably prime)

   Advantages:

- This method is quite good for any value of n, preferably n should be prime.
- It requires a single division operation.
- Very fast.

Disadvantages:

- Consecutive keys maps to consecutive hash values.
- On one hand, consecutive keys do not collide, but on the other hand, it means that consecutive array locations will be occupied which may lead to degradation in performance.

2. Multiplication method

   Step 1: Choose a constant A where $0 < A < 1$.

   Step 2: Multiply the key k by A so that we get kA.

   Step 3: Extract the fractional part of kA so that we get $kA \mod 1$.

   Step 4: Multiply the result obtained at step 3 by the size of the hash table m.

   $h(k) = \lfloor m * (kA \mod 1) \rfloor$

   $n \mod 1 = n - int(n)$

   Knuth suggested the value of $k = \frac{\sqrt{5}-1}{2} = 0.6180339887$

3. Mid-square method

   Step 1: Find the value of the square of the key.

   Step 2: Extract the middle r digits from the result obtained in step 1.

4. Folding method

   Step 1: Divide the key values into n number of parts, say $k_1, k_2, k_3, k_4 \ldots k_n$ where each k_i has the same number of digits except for the right part which may have lesser number of digits than the others.

   Step 2: Obtain $k_1 + k_2 + k_3 + k_4 \ldots k_n$, the hash value is produced by ignoring the last carry (if any).