Hashing

- The key operations in any data processing application, find/membership search, insert, and delete in a collection of data item.
- Organizing data into a table such that it provides 3 key operations mentioned above constitute a dictionary.
- Putting it in analogy with a paper dictionary, we use a key value for extracting an index into the table to reach exact or near-about location for a quick retrieval or an insertion or a deletion of the actual item.
- Binary search tree, heaps, red-black trees, linked lists, other linear lists such as stacks, queues also allow the three basic operation.
- But the performance of these structures vary widely between $\mathrm{O}(n)$ to $\mathrm{O}(\log n)$ for a dictionary operation.

Hashing

- Of course, arranging the data items in an array can realize search, insertion, deletion operations in O(1) time each provided we know the index of the item.
- One limitation of array structure is that we must be aware of data size before-hand to allocate space for an array of required size.
- But array is not suitable when dictionary operations are required for a sample of data items about which we don't have any information such as
  - range of values,
  - number of items,
  - a unused index for inserting a new item.

Hashing

- Which means if the frequency of dictionary operations need to be performed is very high and we don't have any a priori information about the items then using any of above data structures likely to influence the running time to a large extent.
- In such situation the data items are to organized in a way such that all three operations can be done in average O(1) time each.
- Hashing is a technique that uses table based organization to implement the idea. However, no ordering in organization of data items in the table is assumed.

Hashing

- The requirement of O(1) time for search forces that practically no search can be done to search an element!
- So the approach is to extract the index of the item some how in O(1) time.
- Obvious approach is to compute the index or a near-about index in O(1) time.
- The function which is used to compute the index is called a hash function.
- Hash function can operated on just a key associated with the item to produce a table index in O(1) time.
- Therefore, hash tables are known alternatively as associative tables.