Linear Probing without Replacement

Linear probing is a scheme in computer programming for resolving collisions in hash tables, data structures for maintaining a collection of key–value pairs and looking up the value associated with a given key.

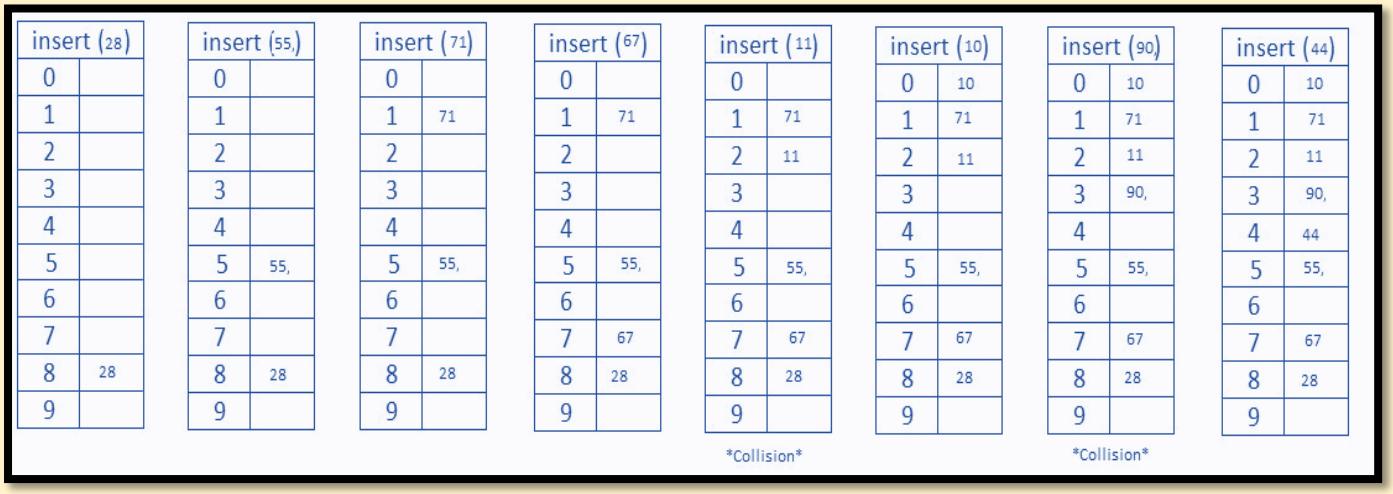
When some data is to be stored in hash table, and if the slot is already occupied by the key then another empty location is searched for a new record. If a location is already occupied then it search for the empty location.

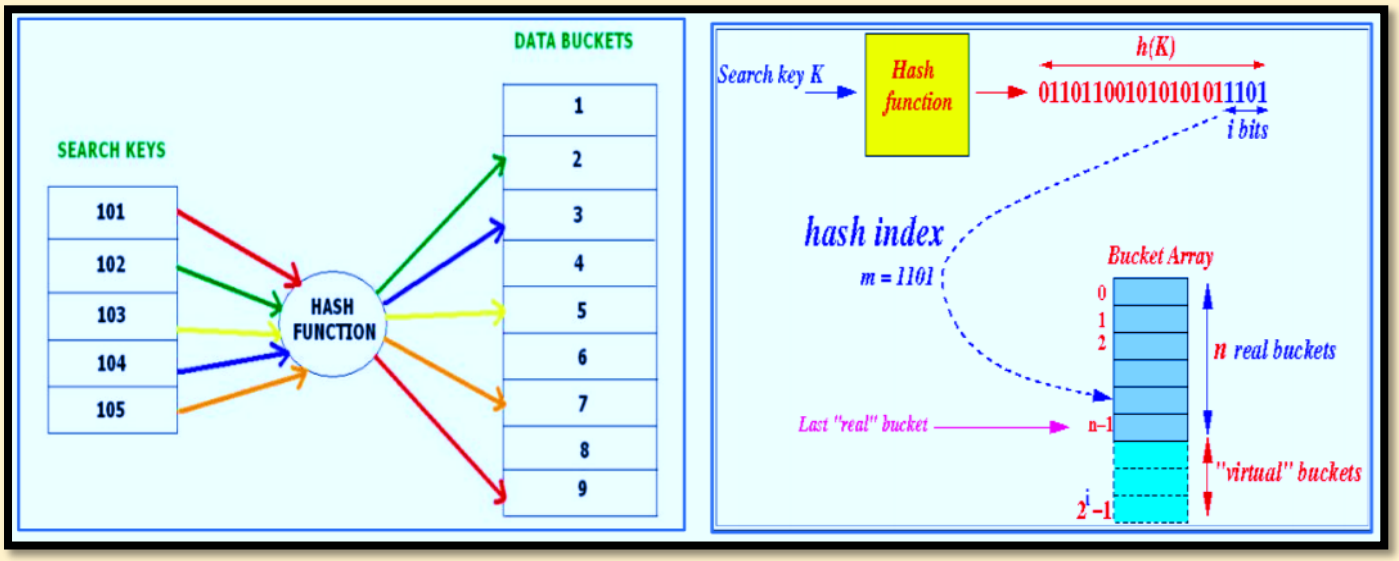
When collision occurs, scan down the array one cell at a time looking for an empty cell.

hi(X) = [(Hash(X) + i) mod TableSize] (i = 0, 1, 2, …)

Compute hash value and increment it until a free cell is found

→Linear Probing without replacement policy: 28, 55, 71, 67, 11, 10, 90 and 44.





Drawbacks:

- Linear probing needs sequentially searching for the next empty cell in the table. But it may take a long time, especially when most keys are in a contiguous region of the table
- The main problem is of clustering. Many consecutive elements form groups. Then, it takes time to search an element or to find an empty bucket.

Roll no	Name	Seat no
226	Sakshi Jagtap	S204148
227	Shruti Dhumne	S204156
228	Mayuri More	S204154
230	Rushikesh Jadhav	S204149