UNIT-II

HASHING WITH LINEAR PROBING

We resolve the collision by adding 1 to the current address.

Assuming that the table is not full

We apply division method of hashing

Consider the example:

0	1	2	3	4	5	6	7
72		18	43	36		6	

Add the keys 10, 5, and 15 to the above table

H(k) = k mod table size

10 mod 8 = 2 a collision, so add 1 to the address then check is it empty or filled. If it is filled then apply the same function, like this we can place this key 10 in the index 5 cell.

0	1	2	3	4	5	6	7
72	15	18	43	36	10	6	5

If the physical end of the table is reached during the linear search will wrap around to the beginning of the table and continue from there.

If an empty slot is not found before reaching the point of collision; the table is full

A problem with the linear probe method is that it is possible for blocks of data to form when collisions are resolved. This is known as **primary clustering**.

This means that any key that hashes into the cluster will require several attempts to resolve the collision.

Linear probes have two advantages: First, They are quite simple to implement. Second, data tend to remain near their home address.

Exercise Example:

Insert the nodes 89, 18, 49, 58, and 69 into a hash table that holds 10 items using the division method.