Extendable hashing example

Subject : Database Systems

Technique which makes hashing dynamic, using both hashing and trie.

Number of buckets grow/shrink or/and hashing uses bit length dynamically as per requirement.

The number on directory hash table/ bucket address table is global depth, indicating current hash prefix. i.e. Bit length based on which partitions are formed.

In book this is referred as 'i'.

The number on bucket is local depth, indicating that for the given bucket, out of global depth how many bits are common values for all keys in given bucket.

In book this is referred as i1, i2, i3 ,... , ij, etc meaning local depth of bucket 1, 2, 3....,j. 'j' is bucket number.

So, if the box numbered i2 on top of bucket#2 contains 5 as data value that means keys inside bucket#2 shall have first five bits common data. i.e. **00000**011, **00000**111

5 here is local depth for bucket#2.

Do not confuse with bucket numbers (j) they are just labels from top to bottom display and used just to mention which bucket in the display the current discussion is about.

Buckets can have different local depth. Whenever, local depth is smaller than global depth means there are more arrows pointing to this bucket. In this scenario whenever new record is required to be entered in this bucket and if it is full, split occur but it does not result in global depth update. Just that new bucket is added, redistribution of existing keys from the bucket and the new one happens based on current local depth+1 bit position. Finally arrows are distributed to these buckets and local depth is increased by one.

1. The key size that maps the directory (the global depth), and
2. The key size that has previously mapped the bucket (the local depth)

1. Doubling the directory when a bucket becomes full
2. Creating a new bucket, and re-distributing the entries between the old and the new bucket

* chaining in the case of overflow and having the same exact key or have reached to max bit length to be used for hashing.

Exercise 11.6 (Sixth Edition, DSC by Silberschatz, Korth, Sudarshan)

Suppose that we are using extendable hashing on a file that contains records with the following search-key values:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31 (10 items)

Additionally 35

Show the extendable hash structure for this file if the hash function is h(x) = x mod 8

Bucket size is 3.

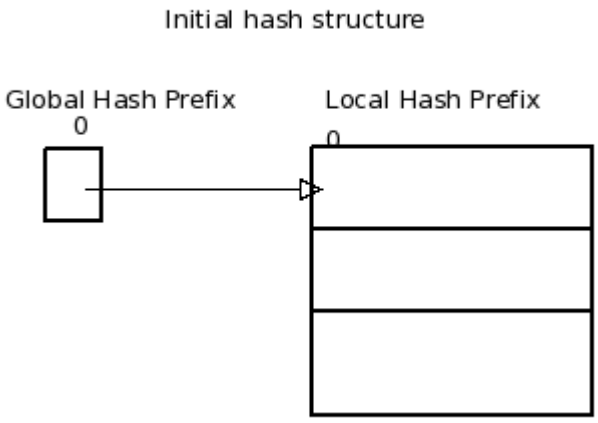
Last page shows complete solution. Try solving yourself and match with last page. If doing first time then follow page by page insertion of records based on their arriaval sr/recno.

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
1	2	2	010
2	3	3	011
3	5	5	101
4	7	7	111
5	11	3	011
6	17	1	001
7	19	3	011
8	23	7	111
9	29	5	101
10	31	7	111

*** All Diagrams, show data in bucket using following notation:**

RecordNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)

Figure (1)



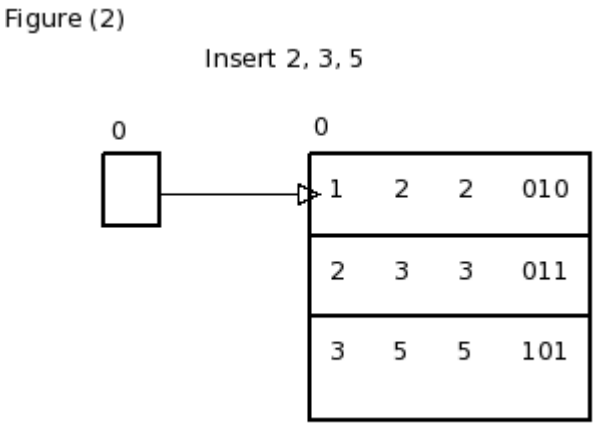
Insert 2, 3, 5

Nothing to do with distribution as global hash prefix is 0. Simply the bucket is available, go ahead and store records as it is not full.

* If look up has to be performed because global hash prefix is 0, pow(2,0) only single bucket is available to search.

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
1	2	2	010
2	3	3	011
3	5	5	101

Figure (2)



Insert 7

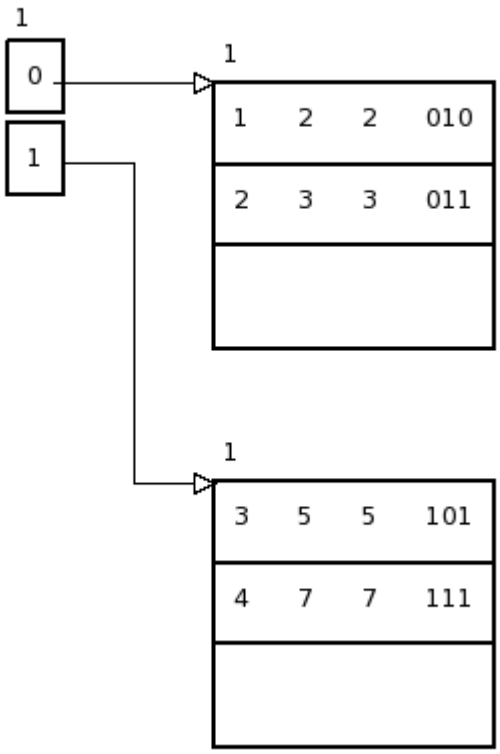
SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
4	7	7	111

Here, bucket is full and the global hash prefix is zero. Hence, lets increase the number of bits that we use from the hash value. We now use 1 bit, allowing us pow(2,1)=2 buckets.

Redistribution of records including new record with bit on position 1 only. Hence, global and local both are set to 1. All records starting with 0 are placed into bucket pointed by address space 0. All records starting with 1 are placed into bucket pointed by address space 1.

Figure (3)

Insert 7

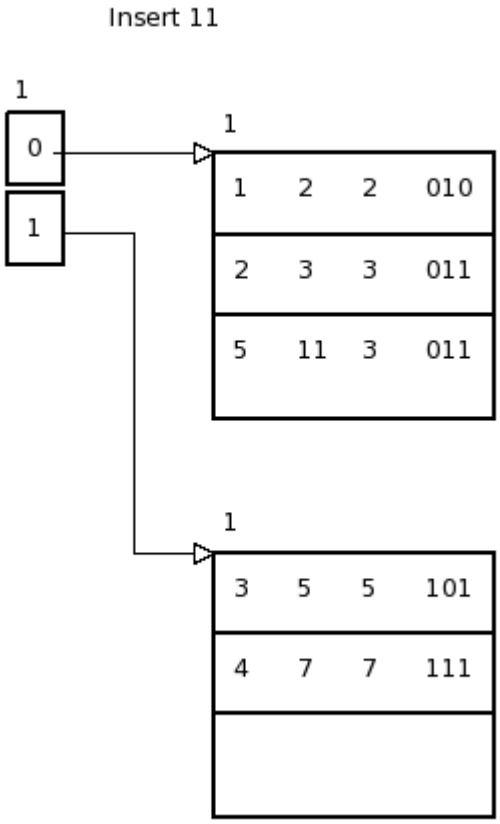


Insert 11

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
5	11	3	011

It goes to bucket pointed by 0 prefix and it has space so right away entry.

Figure (4)



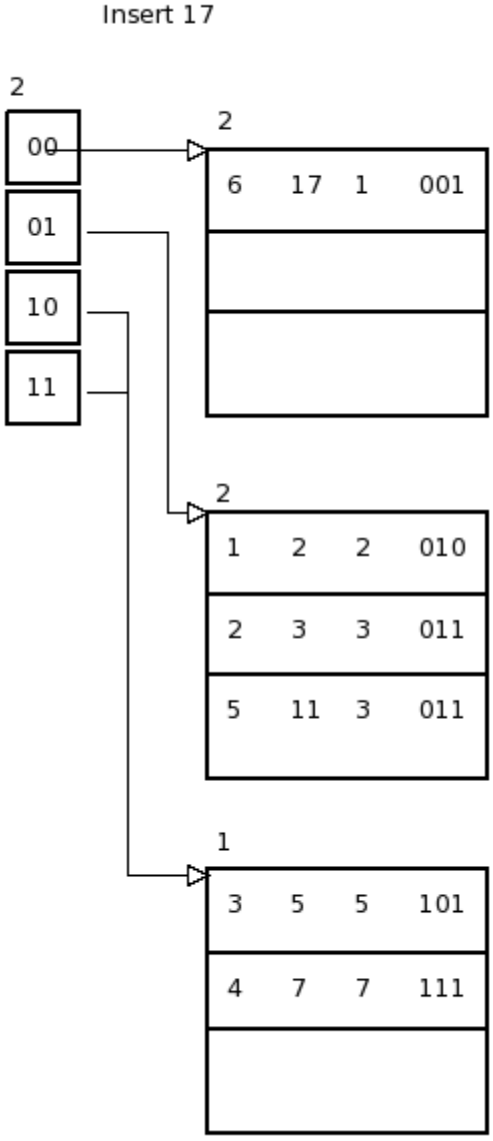
Insert 17

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
6	17	1	001

Here, 17 goes to bucket pointed by 0 but it is full. Again we need to increase the global hash count to accommodate. Note that we have not touched the bin pointed by 1, hence 10 and 11 both points to the same exact bin and the local hash count is still 1 for that bin meaning only first bit is common amongs all record of that bin. And the value is 1.

Redistribution happens of bucket pointed by o to 00 and 01. and now local hash function of both these buckets are 2.

Figure (5)



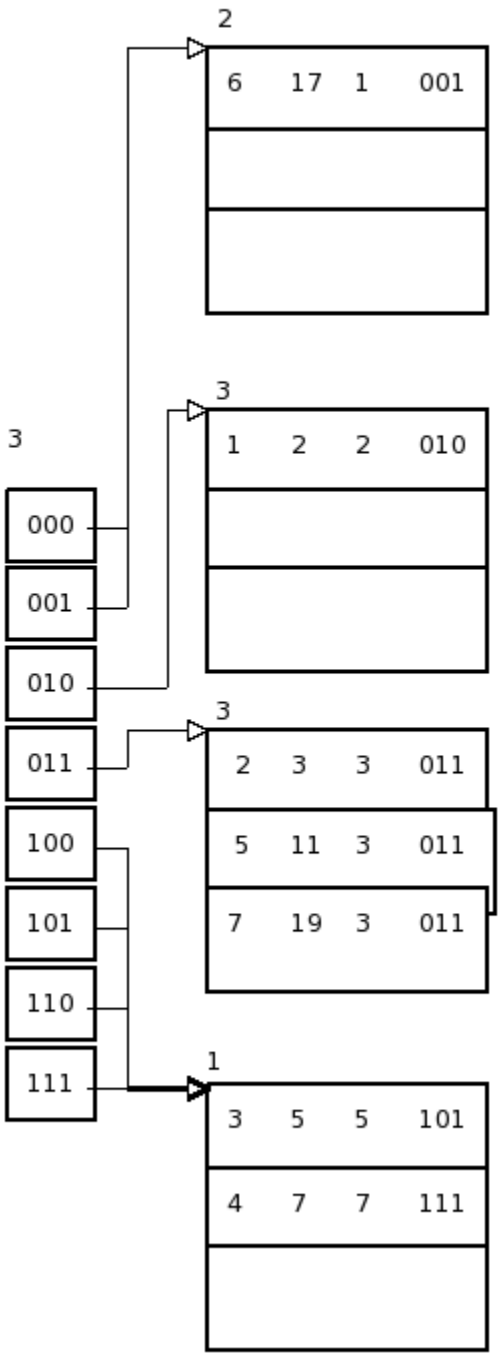
Insert 19:

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)

7	19	3	011

Based on global prefix 2, 01 bucket is full for this new record. Also, local is same as global hence, split with increment to global hash prefix.

Figure (6)

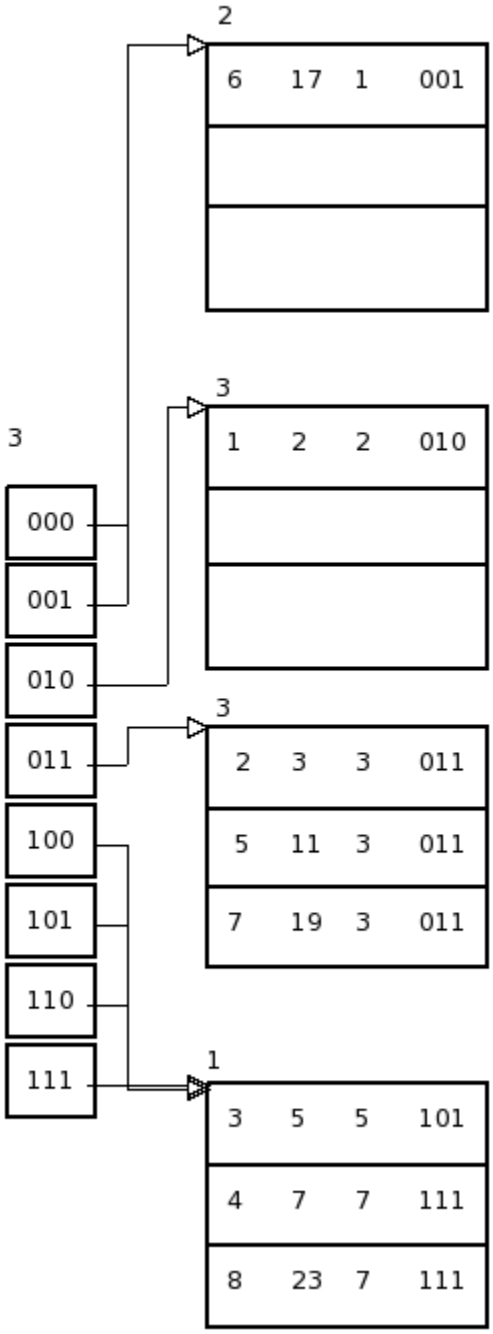


Insert 23

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
8	23	7	111

Here, based on the global count 3, we need place this into bucket pointed by 111. There is space and no adjustment needed.

Figure (7)
Insert 23



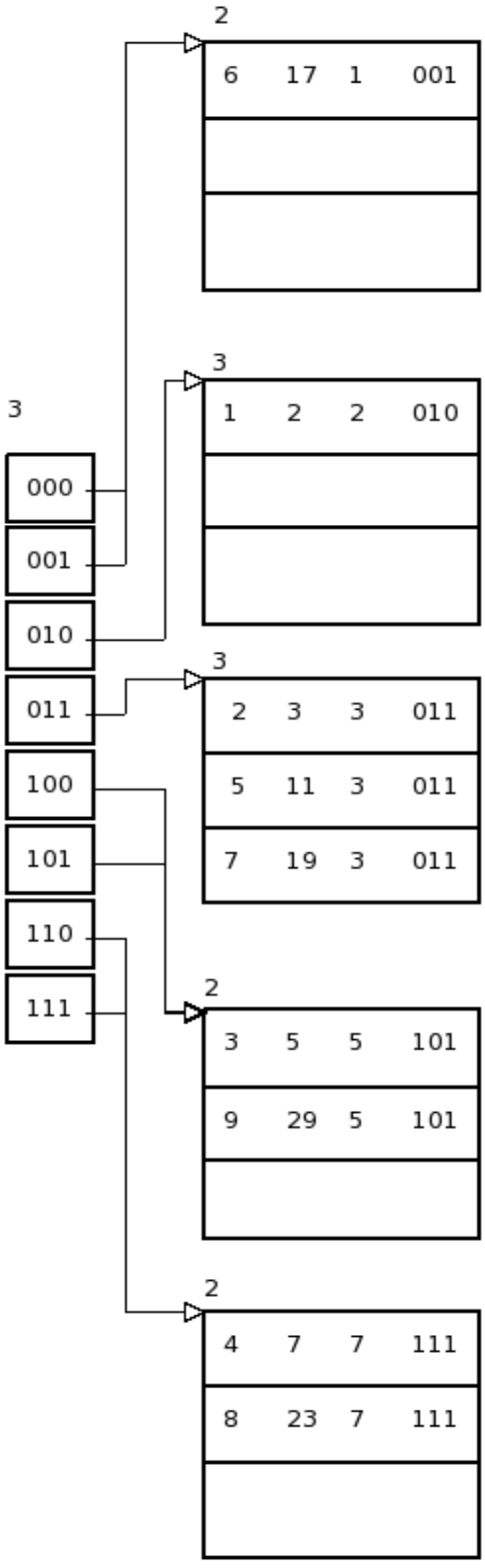
Insert 29

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
9	29	5	101

Here, based on 101, it shall go to bucket pointed by 101. But it is full. Notice that the local prefix is 1, which is smaller than global prefix 3 and hence new bucket will be created and distribution will occur but global prefix need NOT increment. Anyways the hash function limits to 7(111) length 3 in this case though.

Figure(8)

Insert 29



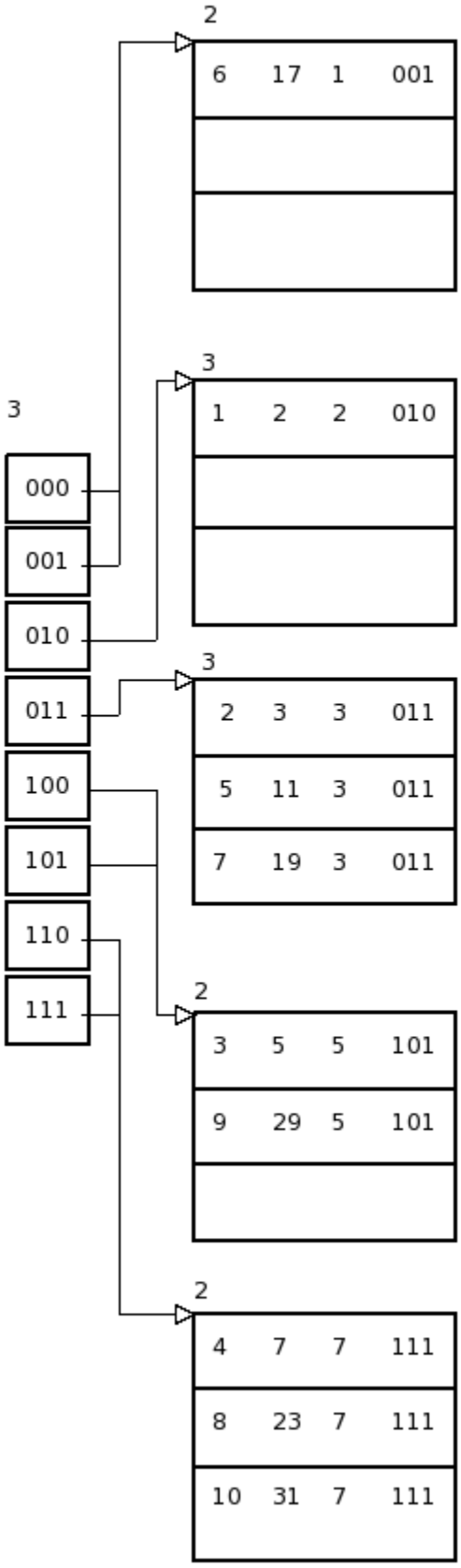
Insert 31

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
10	31	7	111

To insert 31, based on 111, it shall go to bucket pointed by 111 and there is space. So, directly add key record.

Figure(9)

Insert 31



Moreover, lets try to insert 35 now.

SrNo	Data X	h(X)=x mod 8	DEC2BIN(Number;places)
11	35	3	011

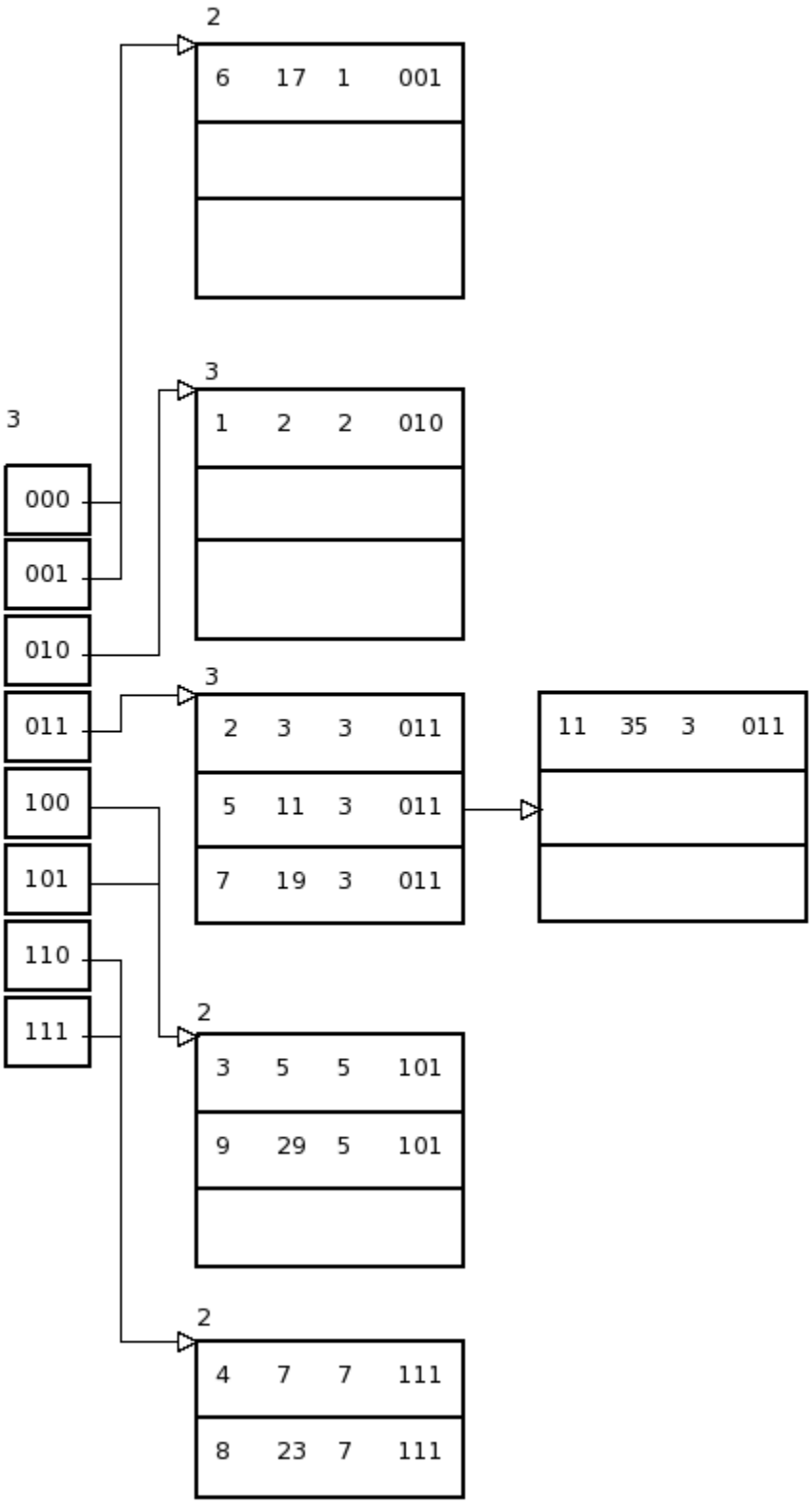
See that the max possible total bits to consider hash function limited to max value 7 which is 111 and the length is 3.

We can not grow our global hash count. Moreover, to insert 35, it shall go to bucket pointed by 011, which if already full. No split is possible because local is same as global same as the max possible.

Lets do chaining here now.

Figure(10)

Insert 35



All in one.
You may see and edit diagram via linux dia diagram editor the files in the folder.