Module – 5 Part B

Extendible Hashing

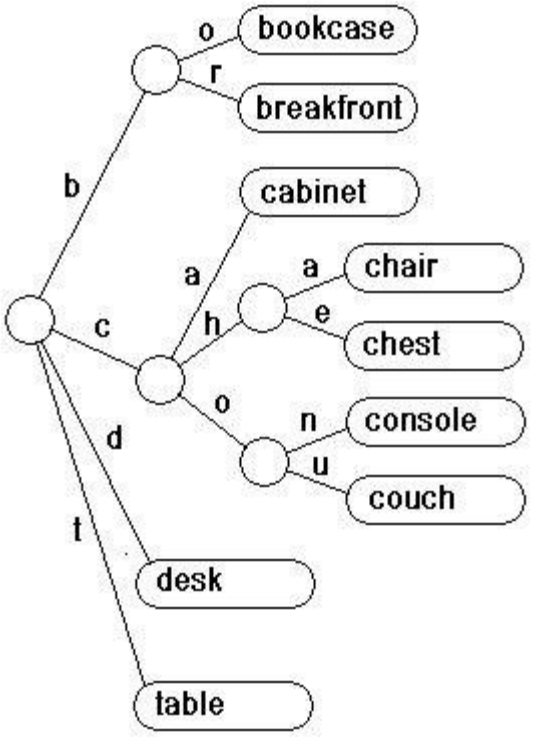
How Extendible Hashing Works

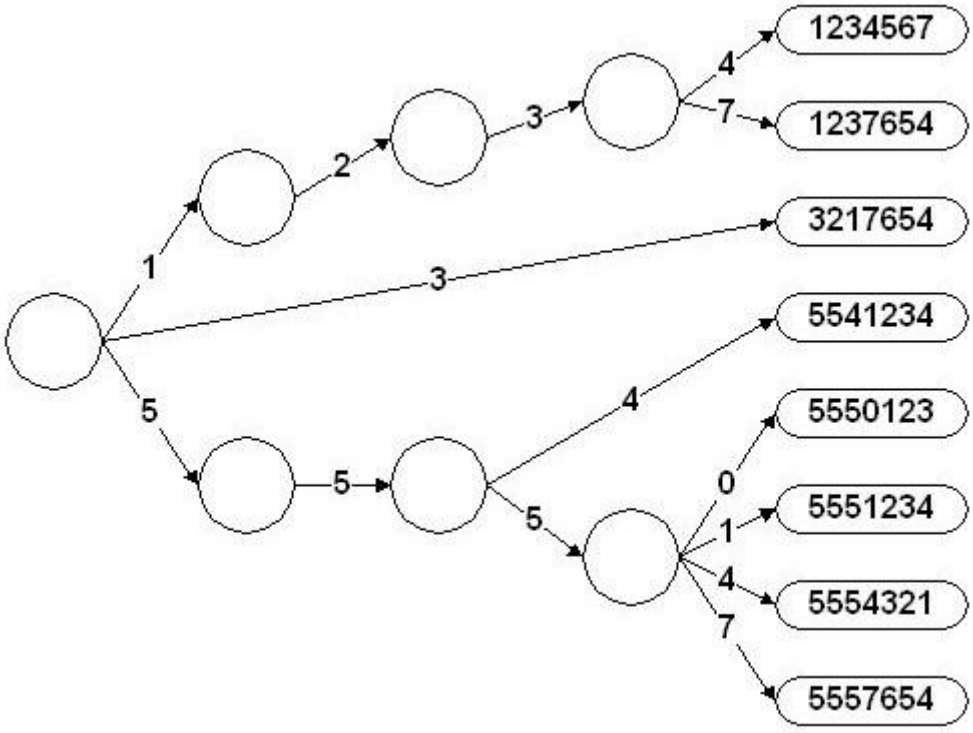
Tries

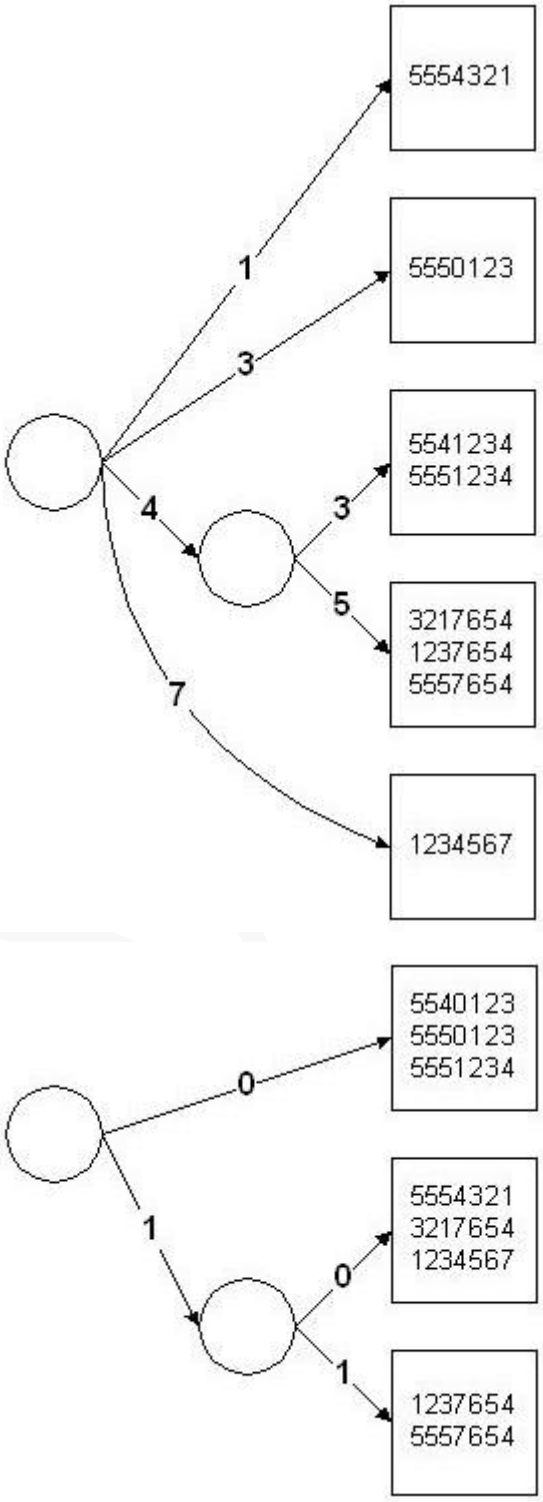
- **trie**

  A search tree in which the child of each node is determined by subsequent charaters of the key.
- An alphabetic (radix 26) trie potentially has one child node for each letter of the alphabet.
- A decimal (radix 10) trie has up to 10 children for each note.
- The trie can be shortened by the use of buckets.
- The bucket distribution can be balanced by the use of hashing.

Key	Hash
5554321	100111001
5550123	10111010
5541234	100111100
5551234	1011110
3217654	100111101
1237654	10011011
5557654	101110011
1234567	1101001


bookcase
breakfront
cabinet
chair
chest
console
couch
desk
table
b
o
r
a
c
h
a
e
o
n
u
d
t

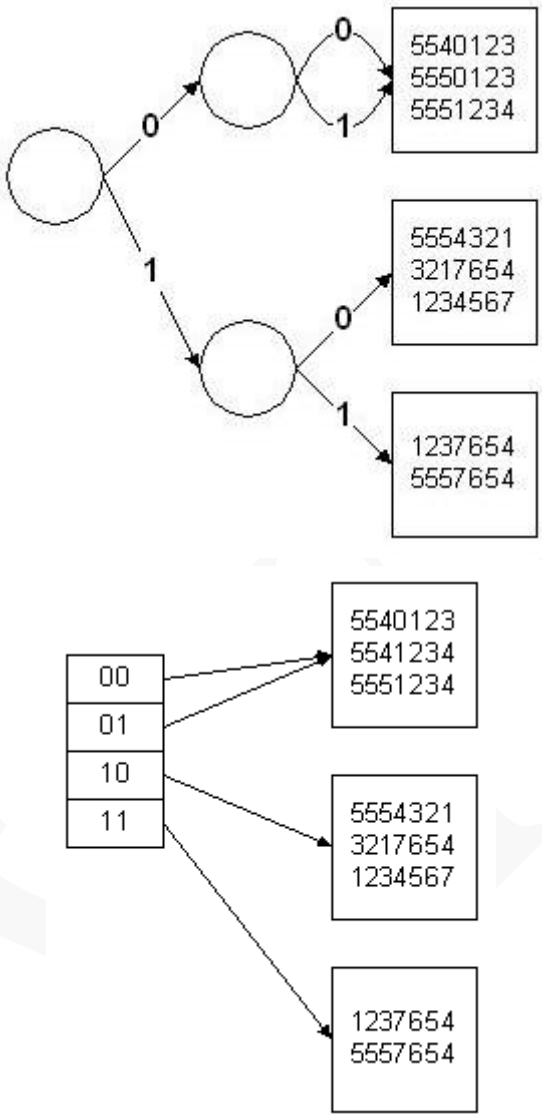

1234567
1237654
3217654
5541234
5550123
5551234
5554321
5557654
1
2
3
4
7
3
5
5
4
5
0
1
4
7



Turning the Tries into a Directory

- **extendible hashing**

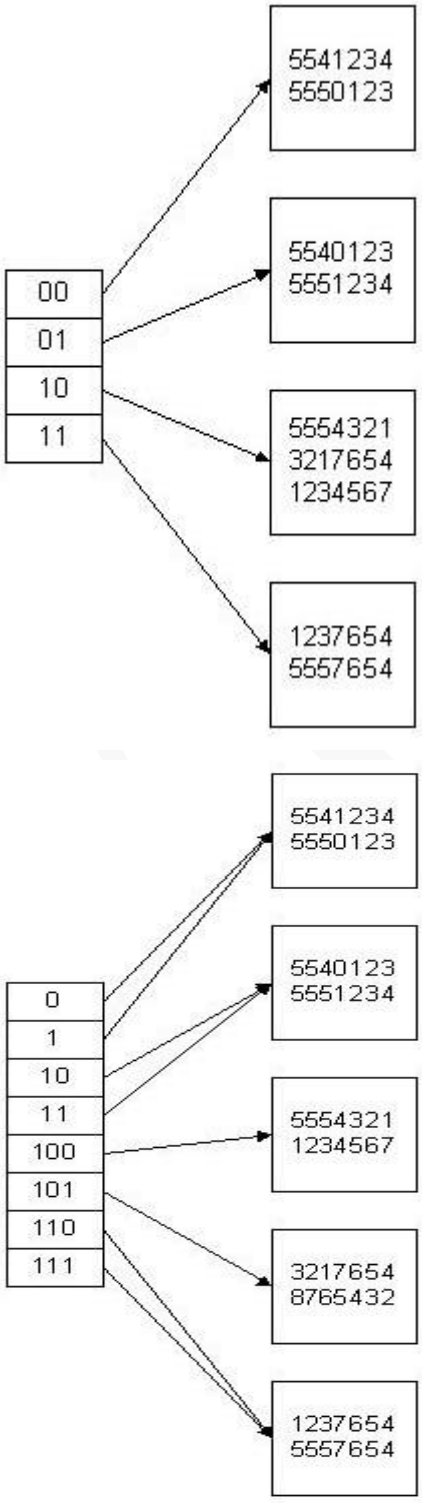
  An application of hashing that works well with files that over time undergo substantial changes in size.



Splitting to Handle Overflow

- **splitting**

  Creation of a new node when a node overflows, with the partial distribution of the contents of the overflowing node to the new node.



Linear Hashing

linear hashing

An application of hashing in which the address space is extended by one bucket each time an overflow occurs