Introduction :-

* Dictionaries :-

A Dictionaries is a collection of Elements; Each Element has a field called key and no two Elements have the same key

* operations performed on a dictionary are

a) Insert an Element with a specified key value

b) Search the dictionary for an element with Specified key value

c) Delete an Element with a specified key value

* Abstract data type of dictionary {

instances

collection of Elements with distinct keys

operations

create() : create an empty dictionary

Search(k,x) : return Element with key k in x ; return false if the operation fails, true if it succeeds

Insert(x) : Insert x into the dictionary

Delete(k,x) : delete element with key k and return it in x

}

* Accessing of Dictionary Elements :-

1) Random Access :- Any Element in the dictionary can be retrieved by performing a search on its key

2) Sequential Access :- Elements are retrieved one by one in ascending order of the key field

*Example for Dictionary :-

collection of students records in a class

→ (key, value) = (Student number, Exam mark)

* Representation of Dictionary :-

There are mainly three methods of representing Dictionary

1) Linear List Representation

2) Skip List Representation

3) Hashing

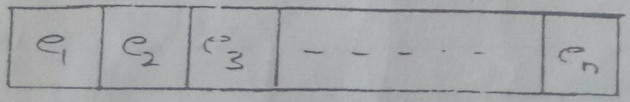
1) Linear List Representation

A dictionary can be represented using linear list representation where the dictionary elements and their keys increases from left to right. The linear list representation again divided into two classes

(i) Sorted List

(ii) Sorted chain

(i) Sorted List :-



* It uses a formula based representation of a linear list

* $L = (e_1, e_2, e_3, ----, e_n)$

Each e_i is a pair (key, value)

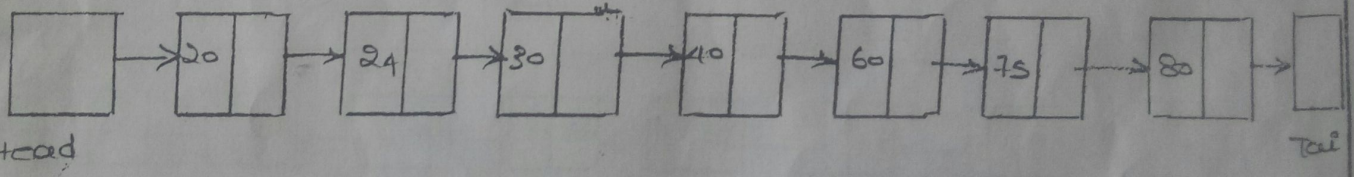
* To search an 'n' Element in the dictionary it takes O(log n) time.

* To make insertion, first verify that the dictionary doesn't already contain an Element with same key. This verification is done by performing Search operation with O(log n) time and next the insertion of Element is done by additional O(n) time as O(n) Elements must be moved to make room for the new Element.

* For deletion operation Searching is done for the Element which is to be deleted and then deletion is performed.

(ii) Sorted chain :-

In this method Each node in the dictionary have two private numbers data and link. The Sorted array is represented as below.



Sorted array is provided by head and tail nodes as shown in the above figure. Head points to the Starting key value of the Element and tail node keep the Element with value larger than that of any other Element.

Eg:- Suppose we need to Search for the Students details whose Reg No is from 9000 to 9100. Then Head points to 9000 and tail points to 9100

In this method to search 'n' element in dictionary chain requires upto 'n' element comparision. This is time consuming method. To overcome this skip list representation of dictionary is implemented.

Skip List Representation

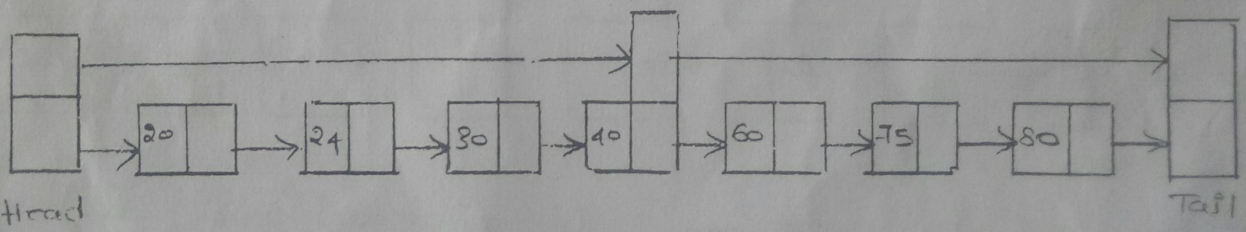
called skip list. Why skip lists:- The linear list with forward pointers is

Although sorted chain takes O(n) time for search of 'n' element, we can improve the search performance of sorted chain by placing additional pointers in some of the chain nodes. These additional pointers permit us to skip over several nodes of the chain during a search operation. Thus it is no longer necessary to examine all chain nodes from left to right during a search.

The different operations performed on skip list are explained by considering below examples.

fig (a)

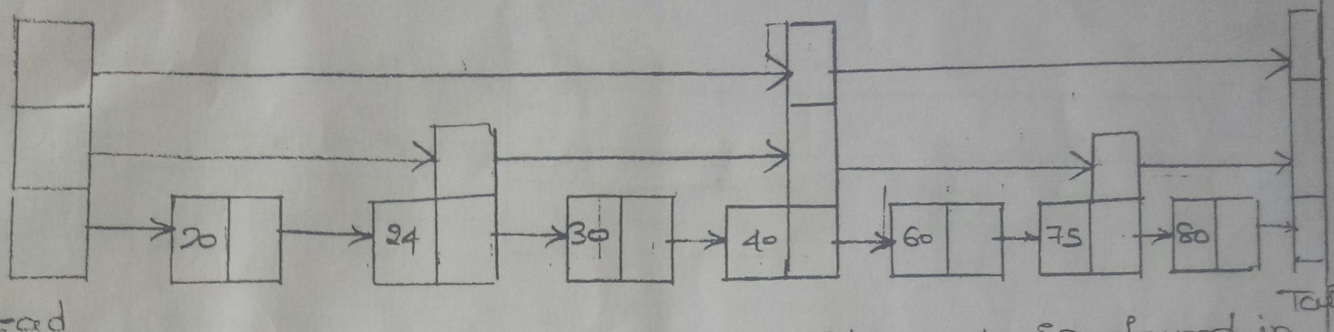
a)



* In this method the number of comparision is reduced to $\frac{n}{2}+1$ by keeping the pointer to the middle element
* If we are looking for the smaller element than the element that pointer pointing, we need not want to search the second half.
* If we are looking for larger element we need comparision of elements only at the right half of the chain

b) We can reduce the number of comparison by keeping pointers to the middle of each half. It is as shown in fig (b)

fig (b)

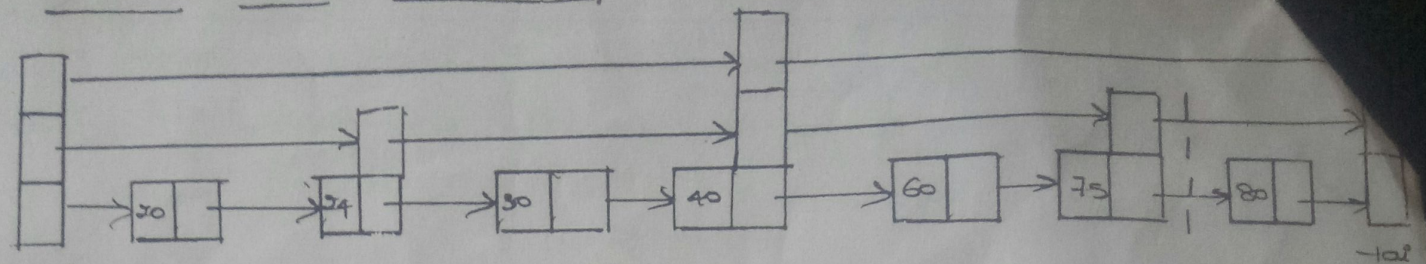


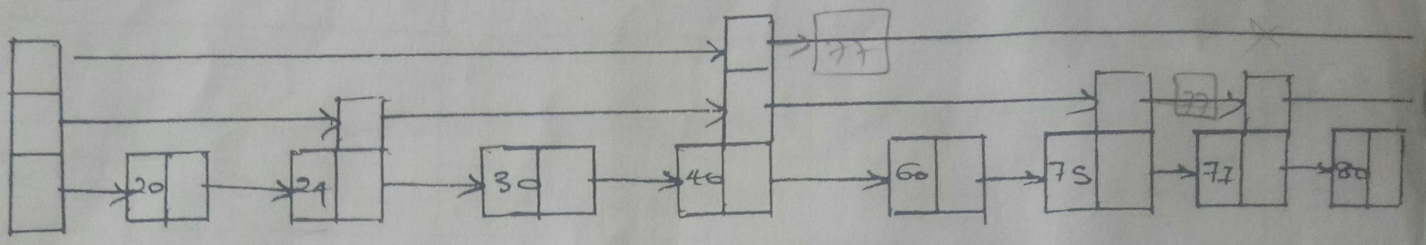
* For example to search 30, the element is found in Θ(1) time using level-2 chain. Since 30<40 the search continues by examing the middle element to left half. This is also done in Θ(1) time using level-1 chain. Since 30>24 the search continues by dropping into the level 0 chain & compares with next element in this chain.

* To search 71, the first comparision is with 40, Since 71>40, drop to level 1 chain and compare with Element 75. 77>75, drop to level 0 chain & compare with the next element i.e, 80 in this chain that comes just after 75. Therefore 77 is not in a dictionary.

* For general n, the level 0 chain includes all elements The level 1 chain includes every second element; the level 2 chain includes every fourth element and the level i chain includes every 2^ith element. The level i chain comprises a subset of elements in the level i-1 chain.

Insertion and Deletion :- fig(a)





fig(b)

To insert an Element with value 77 in the fig(a)

* First Search operation is performed to make Sure no Element with this value is present.
* During this Search the level a pointer is associated with 40, the ~~level pointer~~ is associated with 75 and ~~75~~ is also an Element with level 0. Since 77 is inserted after 75, level and level 1 pointers ~~are~~ get affected as shown in the ~~dotted~~ line.
* To insert an Element we have to assign a level for the newly inserted Element.

Assigning of level :-

* In a regular skip list Structure, a fraction p of the Elements on the level i-1 chain are also on the level i chain. Therefore the probability that an Element is on level i-1 chain is also on the level i chain is 'p'
* Assigning level using Random number generation method

Step 1 :- we have to decide the Maximum level in the dictionary. by choosing the probability

Maximum level, $i = [\log_{1/p} N - 1]$ where N = number of Elements in the dict

Step 2 :- Choose a uniform random number generator which generates random number in the range 0 through Rand_max probability of next random number is ≤ cut-off

where, cutoff = P * Rand_max.

Step 3 :- Generate a random number, If random number generated ≤ cut off, the new element should assign i-1 level

Step 4 :- Now we have to check weather the Element is also in 'i' level. For this generate another random number If the newly generated random number is ≤ cut off then it is also have level i.

Step 5 :- Continue the process untill the random number ≥ cut-off. If the condition is satisfied the Element assign only i-1 level.

In the above example we have to insert 77 at level-1 Element. So we have to satisfy the condition.

Random number generated ≥ cut-off.

* After the insertion of 77 at level '1' Element the SL Linked List structure is as shown in fig(b).

Deletion :- * To delete 77 from the list, we have to do Search operation first. Since the 77 lies at Right half there is no need of control of left.

* The Level 1 and level 0 pointers are associated with 77. By doing so when these pointers are changed to point to the Element after 77 in their respective chains, Structure a fig(a) is obtained. Thus deletion is performed

Disadvantages of skip list :-

1) Skip list is only associated with sorted array.
for unsorted array it cannot be implemented.

2) O(1) performance is not possible

To overcome the above disadvantages the new method of dictionary representation is implemented. The new one is called 'Hashing'