

Skip list

Randomized Data Structures

Priya Bhatia

Agenda of session

- Motivation for Randomized Data Structures
- What and Why Skip List
- Build of a Skip List
- Searching in Skip List
- Insertion in Skip List
- Deletion in Skip List

Motivation for Randomized Data Structures

We've seen many data structures with good **average case** performance on random inputs, but bad behavior on particular inputs

- e.g. Binary Search Trees

□ Instead of randomizing the input (since we cannot!), consider randomizing the data structure

- No bad inputs, just unlucky random numbers
- **Expected case** good behavior on any input

What is a Skip List

- Invented around 1990 by Bill Pugh.
- Generalization of sorted linked lists – so simple to implement.
- Maintains a dynamic set of n elements in $O(\log n)$ time per operation (Searching, Insertion and Deletion).

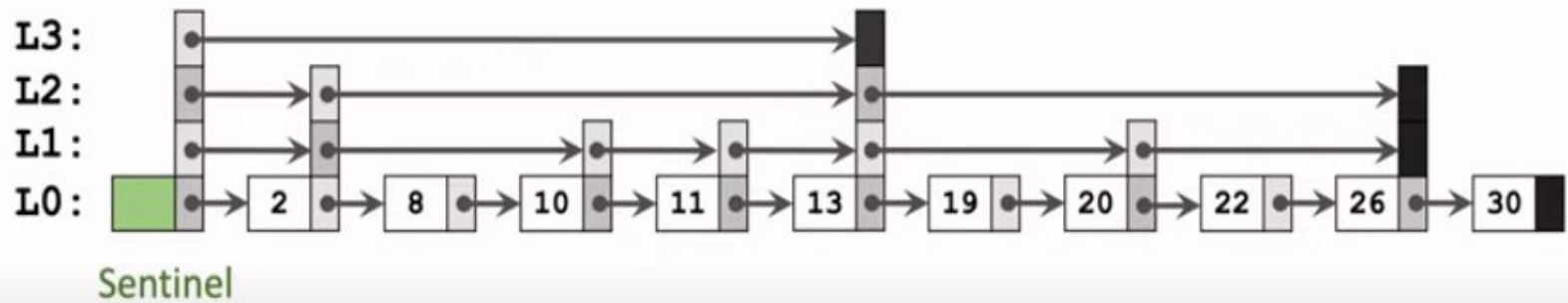
Why Skip List

- Linked List does not support Binary Search even though the list is in sorted fashion and that's the reason of having $O(n)$ time complexity of searching in Linked List.
- Skip List mainly allows faster search and insertion.
- **Search :** $O(\log n)$
- **Insertion :** $O(\log n)$

Build skip list

- New layer is added on top of the bottom list.
- This new layer will include any given element from the previous layer with probability p .
(Event should be unbiased and have each $p = 1/2$).
- For every layer, head node you have to pick up from L_0 to L_n {where number of layers = $\log N$ }.
- For every node, we are making random choices to whether we should grow a height for that particular node or not.
- L_0 layer is fully connected layer and afterwards all above layers are skipping.

Build skip list



Search skip list

- Walk right in top linked list (L1) until going right would go too far.
- Walk down to bottom list(L0) and walk right in L0 until the element is found(or not)
- Suppose key = 8, we want to search
- Start from L3 Layer and we know that key < 13. So, move towards bottom layer which is L2.
- Now, at L2 value of the node is 2 which is less than key, so we move forward and arrive at node 13 and again key < 13, move to bottom layer L1.
- Now, at L1 value is 2 we move forward and got 10 and now key < 10. So, move bottom to L0 layer.

Search skip list

- Now, at L0 we start from 2 and go to 8 and we found out the key.
- Now, at L2 value of the node is 2 which is less than key, so we move forward and arrive at node 13 and again $\text{key} < 13$, move to bottom layer L1.

Insertion skip list

- To insert an element X into a skip list, we need to search(x) to see where X fits into the bottom list (Actual fully connected list).
- Always do the insertion into the bottom list.
- Insert into some of the lists above according to randomization rule.

Deletion skip list

- To delete any element in a skip list, search(x) for that particular element that we want to delete.
- After searching, delete that element from all the levels.

Reference

- Reference Book : Introduction To Algorithms - Cormen