Data Structures and its Applications

Kusuma K V

Department of Computer Science & Engineering

DATA STRUCTURES & ITS APPLICATIONS

UNIT 1: Skip List

Kusuma K V

Department of Computer Science & Engineering

Performance considerations: lists

- Size():
  - O(1) if size is stored explicitly, else O(n)
- IsEmpty():
  - O(1)
- FindElement(k):
  - O(n)
- InsertItem(k, e):
  - O(1) (assumes item already found)
- Remove(k):
  - O(1) (assumes item already found)

Skip List

- Skip Lists support O(log n)
  - Insertion
  - Deletion
  - Search
- A relatively recent data structure
  - W. Pugh in 1989
- “A probabilistic alternative to balanced trees”

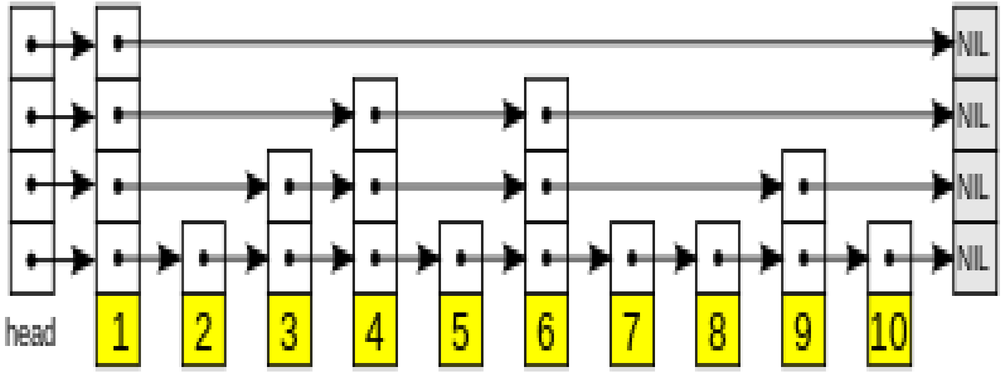
Skip List

- A skip list is a probabilistic data structure that allows O(logn) search complexity as well as O(logn) insertion complexity within an ordered sequence of n elements
- Thus it can get the best features of a sorted array (for searching) while maintaining a linked list-like structure that allows insertion, which is not possible in an array

https://en.wikipedia.org/wiki/Skip_list

Skip List

- Fast search is made possible by maintaining a linked hierarchy of sub sequences, with each successive subsequence skipping over fewer elements than the previous one
- Searching starts in the sparsest subsequence until two consecutive elements have been found, one smaller and one larger than or equal to the element searched for
- Via the linked hierarchy, these two elements link to elements of the next sparsest subsequence, where searching is continued until finally we are searching in the full sequence



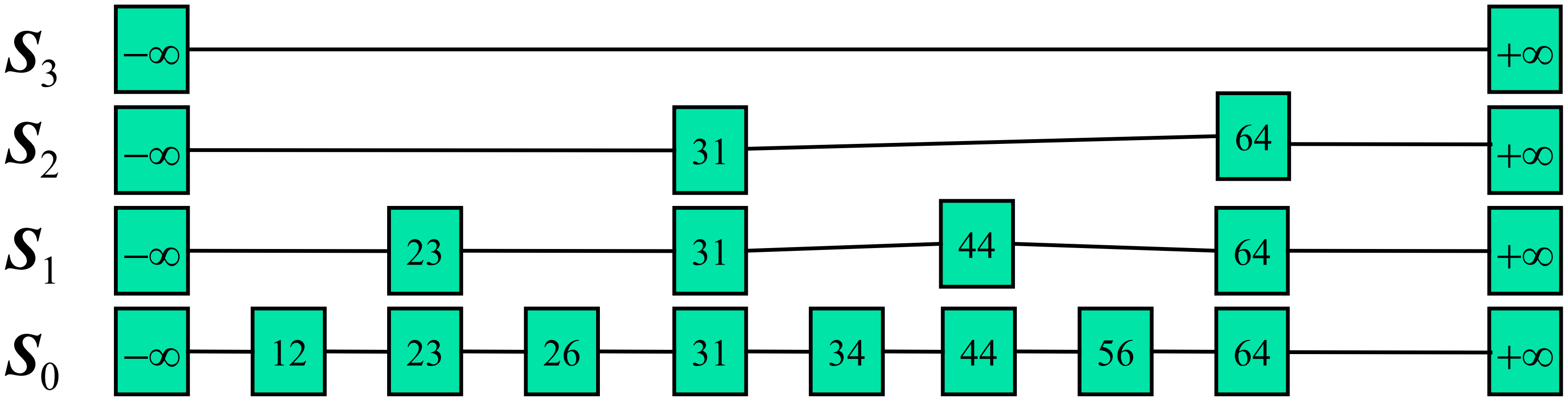
https://en.wikipedia.org/wiki/Skip_list

Perfect Skip List

- A skip list is a collection of lists at different levels
- The lowest level (0) is a sorted, singly linked list of all nodes
- The first level (1) links alternate nodes
- The second level (2) links every fourth node
- In general, level i links every 2^ith node
- In total, $\lceil \log_2 n \rceil$ levels (i.e. $O(\log_2 n)$ levels)
- Each level has half the nodes of the one below it

Perfect Skip List

- Example of a Perfect Skip List



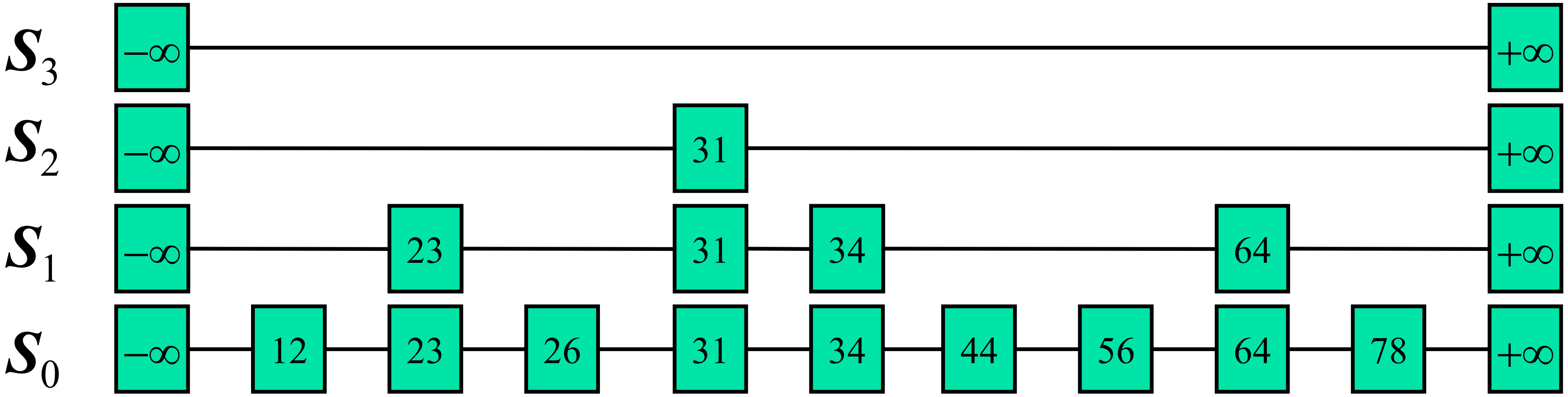
Insertions/Deletions

When we add a new node, our beautifully precise structure might become invalid

- We may have to change the level of every node
- One option is to move all the elements around
- But it takes O(n) time, which is back where we began
- Is it possible to achieve a net gain?

Randomized Skip List

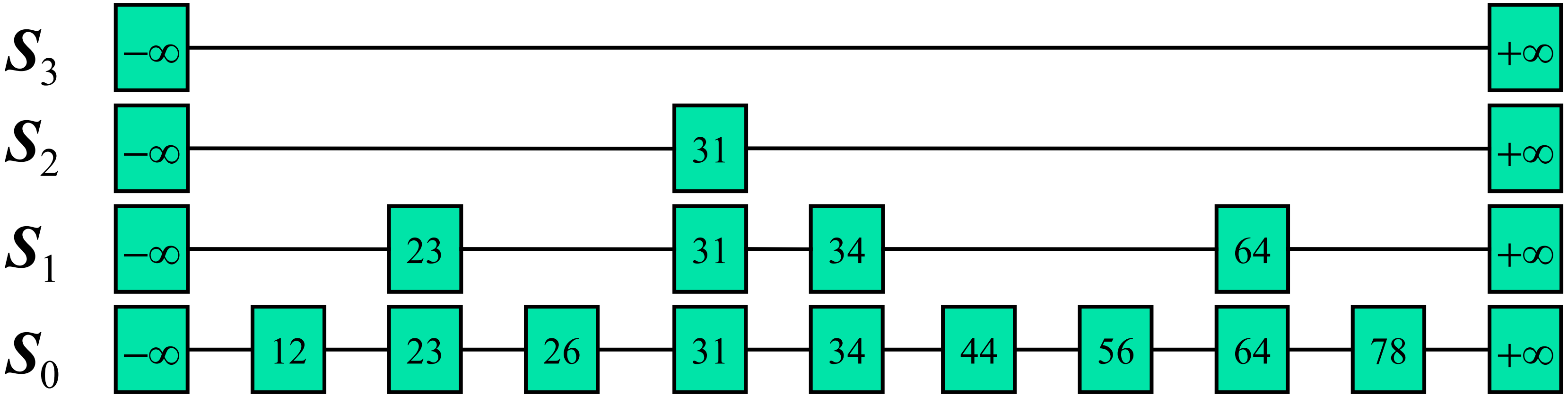
Example of a Randomized Skip List



Skip List

Skip List definition

- A skip list for a set ***S*** of distinct (key, element) items is a series of lists S_0, S_1 , … , S_h such that:
  - Each list S_i contains the special keys +∞ and −∞
  - List S_0 contains the keys of ***S*** in non decreasing order
  - Each list is a subsequence of the previous one, i.e., $$S_0 \supseteq S_1 \supseteq \dots \supseteq S_h$$
  - List S_h contains only the two special keys

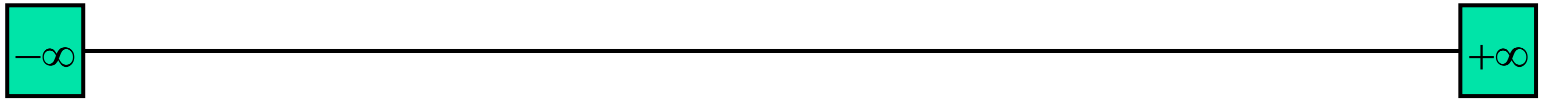


Skip List

Initialization

A new list is initialized as follows:

- A node NIL (+∞) is created and its key is set to a value greater than the greatest key that could possibly used in the list
- Another node NIL (–∞) is created, value set to lowest key that could be used



References

- Presentation Slides: Advanced Data Structures
  Dr. RamaMoorthy Srinath, Professor, CSE, PESU

Note: Apart from the links already mentioned on Slides

1. What is the main advantage of using a skip list over a balanced binary search tree?

a) Skip lists require less memory.

b) Skip lists are easier to implement and provide probabilistic balancing without complex rotations.

c) Skip lists always guarantee O(log n) performance without exceptions.

d) Skip lists can store duplicate elements but BSTs cannot.

Multiple-Choice-Questions (MCQ’s)

1. **What is the main advantage of using a skip list over a balanced binary search tree?**

a) Skip lists require less memory.

b) Skip lists are easier to implement and provide probabilistic balancing without complex rotations.

c) Skip lists always guarantee O(log n) performance without exceptions.

d) Skip lists can store duplicate elements but BSTs cannot.

2. What is the role of randomization in a skip list?

a) It determines the position of nodes in the base linked list.

b) It decides the height of the towers (levels) for each node.

c) It avoids duplicate elements automatically.

d) It ensures all keys are evenly spaced.

2. What is the role of randomization in a skip list?

a) It determines the position of nodes in the base linked list.

b) It decides the height of the towers (levels) for each node.

c) It avoids duplicate elements automatically.

d) It ensures all keys are evenly spaced.

3. Skip lists can be viewed as a combination of which two data structures?

a) Linked List and Hash Table

b) Linked List and Binary Tree

c) Multiple Linked Lists stacked in levels

d) Doubly Linked List and Heap

3. Skip lists can be viewed as a combination of which two data structures?

a) Linked List and Hash Table

b) Linked List and Binary Tree

c) Multiple Linked Lists stacked in levels

d) Doubly Linked List and Heap

4. Which of the following applications is best suited for skip lists?

a) Implementing memory allocation.

b) Designing a concurrent, lock-free, ordered data structure.

c) Solving shortest path problems in graphs.

d) Replacing arrays for random access.

DATA STRUCTURES & ITS APPLICATIONS

Multiple-Choice-Questions (MCQ’s)

4. Which of the following applications is best suited for skip lists?

a) Implementing memory allocation.

b) Designing a concurrent, lock-free, ordered data structure.

c) Solving shortest path problems in graphs.

d) Replacing arrays for random access.

7. Which of the following statements is false about skip lists?

a) Skip lists use multiple levels of linked lists to improve search performance.

b) The bottom level is a sorted linked list containing all elements.

c) Skip lists guarantee O(1) search in all cases.

d) The height of a skip list grows logarithmically with the number of elements (on average).

7. Which of the following statements is false about skip lists?

a) Skip lists use multiple levels of linked lists to improve search performance.

b) The bottom level is a sorted linked list containing all elements.

c) Skip lists guarantee O(1) search in all cases.

d) The height of a skip list grows logarithmically with the number of elements (on average).

THANK YOU

Kusuma K V

Department of Computer Science & Engineering

kusumakv@pes.edu