Data Structures & 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

Skip List

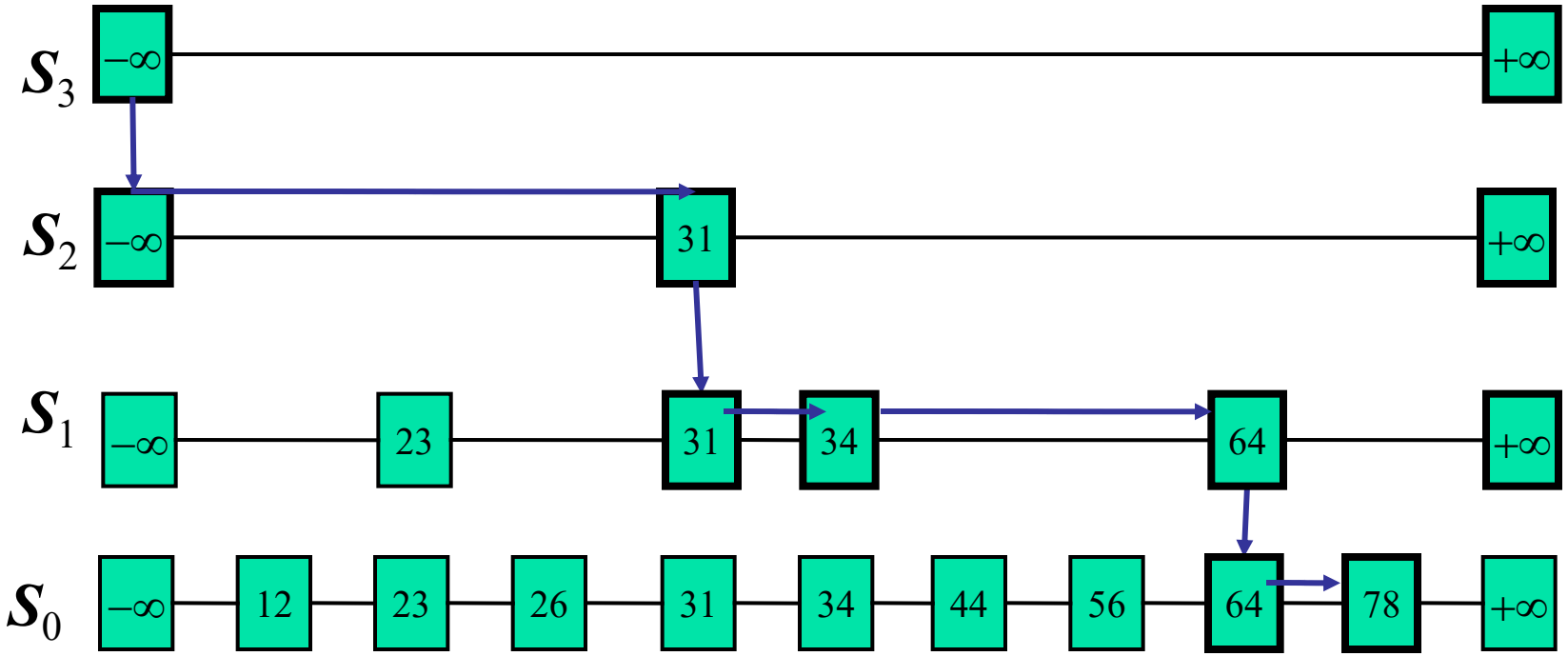
Search

We search for a key x in a skip list as follows:

- We start at the first position of the top list
- At the current position p, we compare x with y i.e., key(after(p))
  - x = y: we return element(after(p))
  - x > y: we “scan forward”
  - x < y: we “drop down”
- If we try to drop down past the bottom list, we return NO_SUCH_KEY
- Example: search for 78

Skip List

Searching in Skip List Example



Skip List

Insertion

- The insertion algorithm for skip lists uses randomization to decide how many references to the new item should be added to the skip list
- We then insert the new item in this bottom-level list at its appropriate position. After inserting the new item at this level we “flip a coin”
  - If the flip comes up tails, then we stop right there.
  - If the flip comes up heads, we move to next higher level and insert in this level at the appropriate position

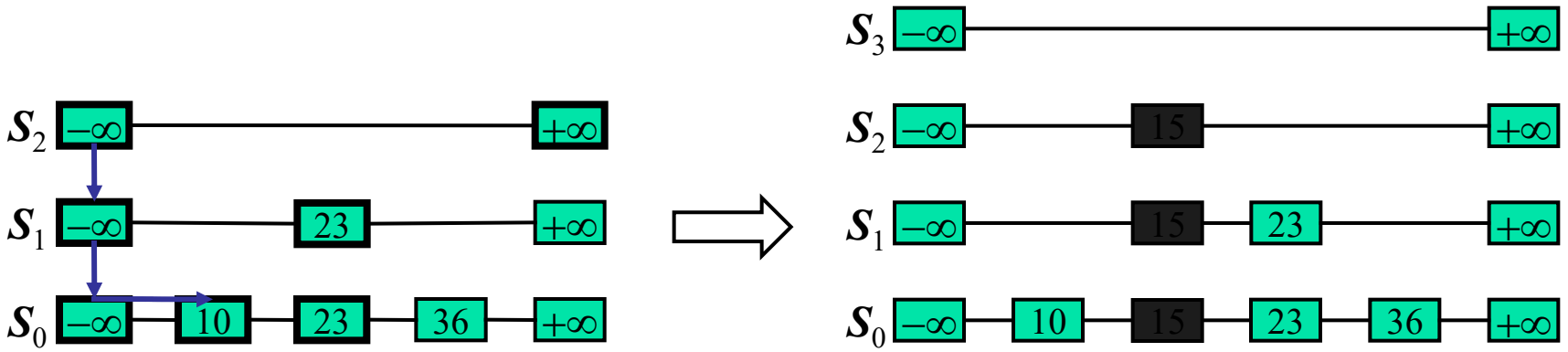
Randomized Algorithms

- A randomized algorithm performs coin tosses (i.e., uses random bits) to control its execution
- Its running time depends on the outcomes of the coin tosses
- We analyze the expected running time of a randomized algorithm under the following assumptions
  - the coins are unbiased, and
  - the coin tosses are independent
- The worst-case running time of a randomized algorithm is large but has very low probability (e.g., it occurs when all the coin tosses give “heads”)

Skip List

Insertion in Skip List Example

- Suppose we want to insert 15
- Do a search, and find the spot between 10 and 23
- Suppose the coin comes up “head” two times



Skip List

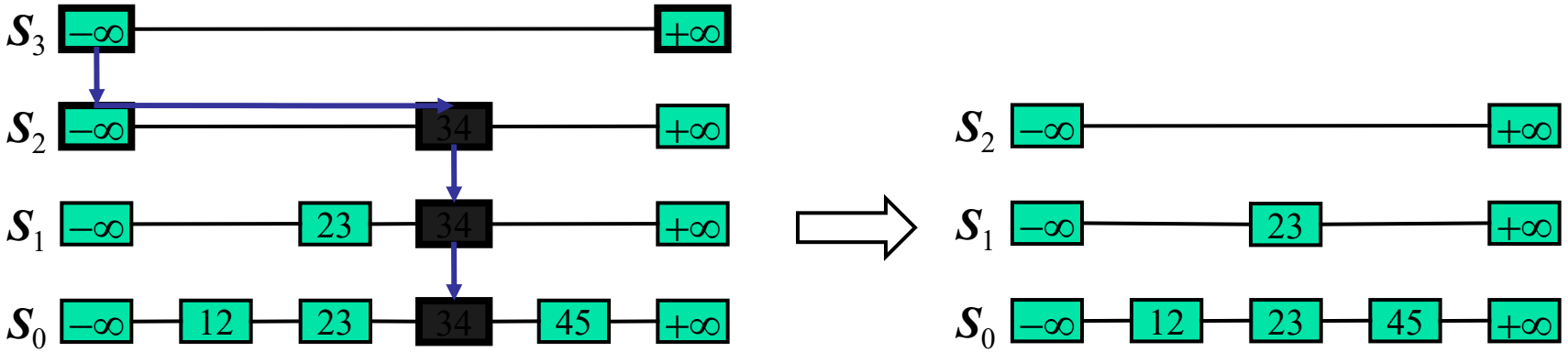
Deletion

- We begin by performing a search for the given key k.
- If a position p with key k is not found, then we return the NO_SUCH_KEY element
- Otherwise, if a position p with key k is found (it would be found on the bottom level), then we remove all the position above p
- If more than one upper level is empty, remove it

Skip List

Deletion in Skip List Example

- Suppose we want to delete 34
- Do a search, find the spot between 23 and 45
- Remove all the position above p



Skip List

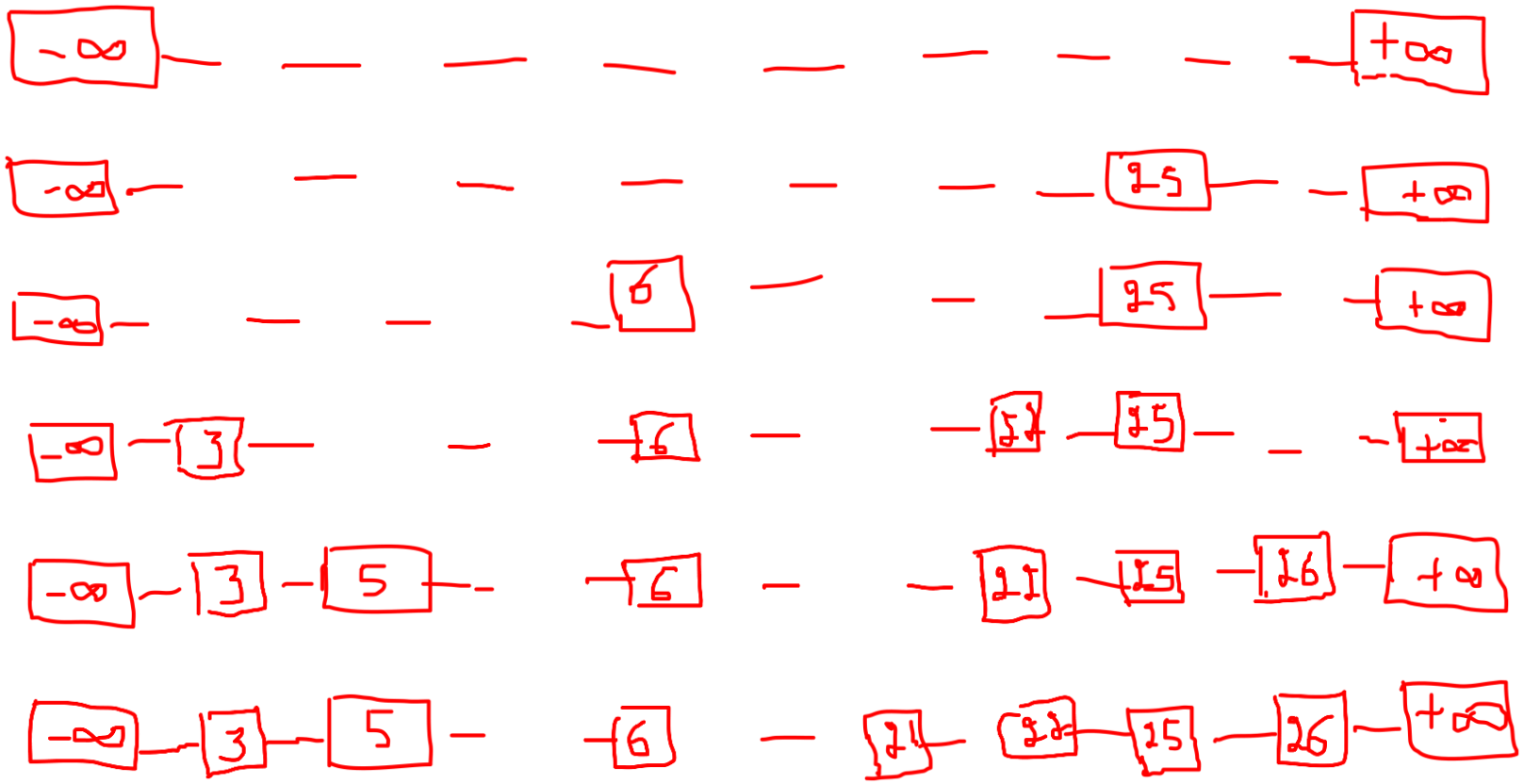
Starting with an empty skip list, insert these keys, with these "randomly generated" levels

- 5, with level 1
- 26, with level 1
- 25, with level 4
- 6, with level 3
- 21, with level 0
- 3, with level 2
- 22, with level 2

Note that the ordering of the keys in the skip list is determined by the value of the keys only; the levels of the nodes are determined by the random number generator only

Skip List

5 with level 1, 26 with level 1, 25 with level 4, 6 with level 3
21 with level 0, 3 with level 2, 22 with level 2



References

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

Note: Apart from the links already mentioned on Slides

1. **While searching for key K in a skip list, what is the correct rule for horizontal movement at the current level?**

a) Move right while next.key <= K.

b) Move right while next.key < K; drop down when next.key >= K or next == NULL.

c) Always move right exactly once per level.

d) Drop down first, then move right to find K.

1. **While searching for key K in a skip list, what is the correct rule for horizontal movement at the current level?**

a) Move right while next.key <= K.

b) Move right while next.key < K; drop down when next.key >= K or next == NULL.

c) Always move right exactly once per level.

d) Drop down first, then move right to find K.

2. Which sequence best describes inserting a new key K?

a) Random level → Allocate node → Search path → Link node in found level only.

b) Search path (collect update[]) → Random level for node → Allocate node → Splice node into all levels ≤ nodeLevel.

c) Allocate node → Splice at level 0 → Randomly add higher links later.

d) Search path → Allocate node → Splice only at top level.

2. Which sequence best describes inserting a new key K?

a) Random level → Allocate node → Search path → Link node in found level only.

b) Search path (collect update[]) → Random level for node → Allocate node → Splice node into all levels ≤ nodeLevel.

c) Allocate node → Splice at level 0 → Randomly add higher links later.

d) Search path → Allocate node → Splice only at top level.

3. How is the level (height) of a newly inserted node typically determined?

a) Fixed at 1 for all nodes.

b) Count set bits in key value.

c) Repeatedly "flip a coin" with probability p of going up a level until fail or max level reached.

d) Proportional to current list size ($\lceil \log_2 n \rceil$).

3. How is the level (height) of a newly inserted node typically determined?

a) Fixed at 1 for all nodes.

b) Count set bits in key value.

c) Repeatedly "flip a coin" with probability p of going up a level until fail or max level reached.

d) Proportional to current list size ($\lceil \log_2 n \rceil$).

4. To delete key K, which approach is correct?

a) Search only level 0; unlink node; done.

b) Search from top without storing predecessors; when found, drop to level 0 and delete.

c) Search building update[] of last nodes < K at each level; if found, relink update[i]→forward[i] to skip target at all involved levels; free node.

d) Mark node deleted; let garbage collector collapse levels.

4. To delete key K, which approach is correct?

a) Search only level 0; unlink node; done.

b) Search from top without storing predecessors; when found, drop to level 0 and delete.

c) Search building update[] of last nodes < K at each level; if found, relink update[i]→forward[i] to skip target at all involved levels; free node.

d) Mark node deleted; let garbage collector collapse levels.

5. After deleting a node, when (if ever) should the current highest level of the skip list be reduced?

a) Never; levels are fixed after first insertion.

b) Whenever any node is deleted.

c) When the header's forward pointer at the top level becomes NULL (no elements at that level).

d) When current size < previous size/2.

5. After deleting a node, when (if ever) should the current highest level of the skip list be reduced?

a) Never; levels are fixed after first insertion.

b) Whenever any node is deleted.

c) When the header's forward pointer at the top level becomes NULL (no elements at that level).

d) When current size < previous size/2.

THANK YOU

Kusuma K V

Department of Computer Science & Engineering

kusumakv@pes.edu