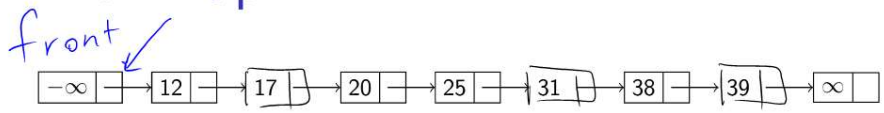
I & C SCI 46 Winter 2023
Lecture 7: Skip Lists

2

Warm Up



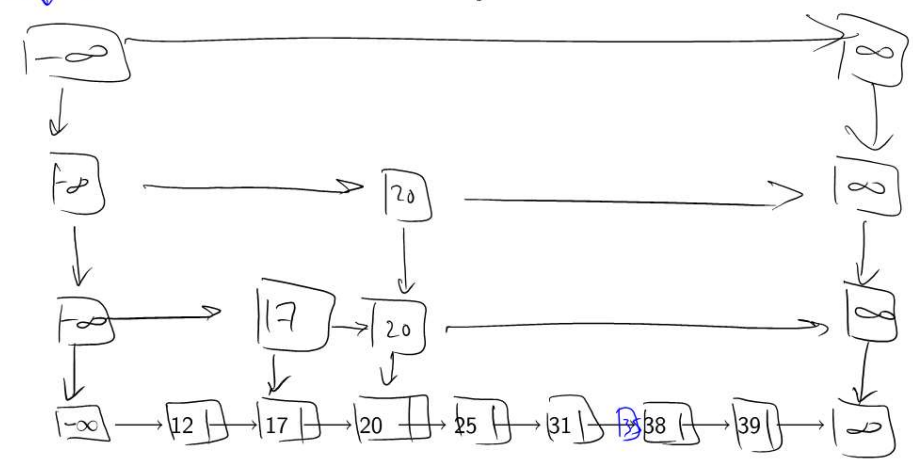
- Linked list is sorted
- Linked list has $-\infty$ and ∞
- Want largest key whose value is *at most* k.

```
find(k)
    t ← front   // t->key ≤ k always
    while (t -> next -> key ≤ k)
        t = t -> next
    return t
```

3

Improved running time

- Add a layer. Does it help?
- Which get added to higher layer?
- How about another layer?



4

New Find Function

```
old-find(k) // for reference
  t ← L.first
  while t.next.key ≤ k do
    t ← t.next
  return t.key
```

How does find change for layered list?

```
t ← top-left // invariant: t->key ≤ k
while (t -> down ≠ nullptr)
{     t = t -> down
      while (t -> next -> key ≤ k)
            t = t -> next
}
return ?
```

5

Inserting into a Skip List

`Insert`(k, v)

previous to new

under new

// assume we have insertAfterAbove(p, q, k, v)

// and no new levels needed

returns the new node

```
p = find(k)
q = iAA(p, nullptr, k, v)
while coinflip() is heads.
    while p->up is nullptr
        p = p->left
    p = p->up
    q = iAA(p, q, k, v)
```

6

Running time for find

- Suppose skip list has height h
- How long does a find take?

- We added key. Probability stored i levels up?

$$\left(\frac{1}{2^i}\right)$$

7

Height of a skip list

- Insert n keys to initially empty list
- P_i = prob level i has at least one item?

$$P_i \le \frac{n}{2^i}$$

- Probability height at least $\underbrace{3\log n}_{i}$?

$$P \le \frac{n}{2^{3\log n}} = \frac{n}{(\cancel{2^{\log n}}\ n)^3} = \frac{1}{n^2}$$

8

Time for find?

```
find(k)
  t ← topmost left node of list
  while t.below ≠ nullptr do
    t ← t.below
    while t.after.key ≤ k do
      t ← t.after
  return t
```

$O(\log n)$

- How many drops?

height

50% each

- How many scan forward?

9

Size of a skip list

- E[# items] at level i is $n/2^i$

$\leq 2n$