Warm Up

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

```plaintext
find(k)

$\text{t} \leftarrow \text{first}$

while ($\text{t} \rightarrow \text{next} \rightarrow \text{key} \leq k$)

$\text{t} = \text{t} \rightarrow \text{next}$

return $\text{t}$
```
Improved running time

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

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 // row only -∞, ∞
    while (t.down ≠ nullptr)
        while (t.next.key ≤ k)
            t = t.next
    return t
Inserting into a Skip List

Insert\((k, v)\)
// assume we have insertAfterAbove\((p, q, k, v)\)
// and no new levels needed
\[ p = \text{find}(k) \]
\[ q = i\text{AA}(p, \text{nullptr}, k, v) \]
While (com::flip() is heads)
\[ p = p \rightarrow \text{left} \]
\[ p = p \rightarrow \text{up} \]
\[ q = i\text{AA}(p, q, k, v) \]

Running time for find

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

- We added key. Probability stored \( i \) levels up?
\[
\frac{1}{2^i}
\]
Height of a skip list

- Insert $n$ keys to initially empty list
- $P_i = \text{prob level } i \text{ has at least one item}$?

\[
P_i \leq n \times \frac{1}{2^i}
\]

- Probability height at least $3 \log n$?

\[
\leq \frac{n}{2^{3 \log n}} = \frac{n}{(2^{\log n})^3} = \frac{n}{n^3} = \frac{1}{n^2}
\]

Time for find?

```plaintext
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
```

- How many drops?

  half of nodes visited, $O(\log n)$

- How many scan forward?
Size of a skip list

- $E[\# \text{ items}]$ at level $i$ is $n/2^i$

\[ \sum \leq 2n \]