SKip Lists

- invented by Bill Pugh c. 1990
- implements ADT Dictionary (insert, delete, search by key value)

Idea

- For Dictionary, Arrays + Linked Lists have different benefits / drawbacks.
- would be nice to have best of both.

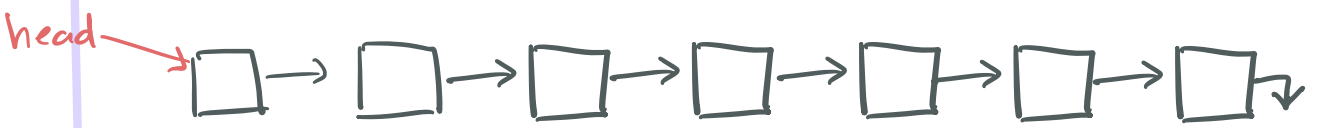
(Packed) Sorted Array



- great for searching because Random Access $O(\lg n)$

- not so great for insertion, deletion $O(n)$

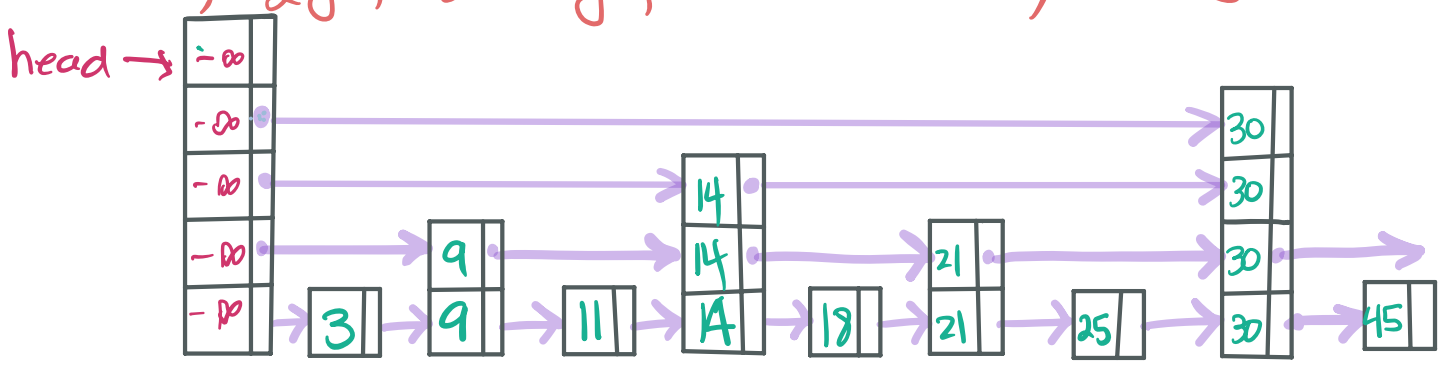
Sorted Linked List



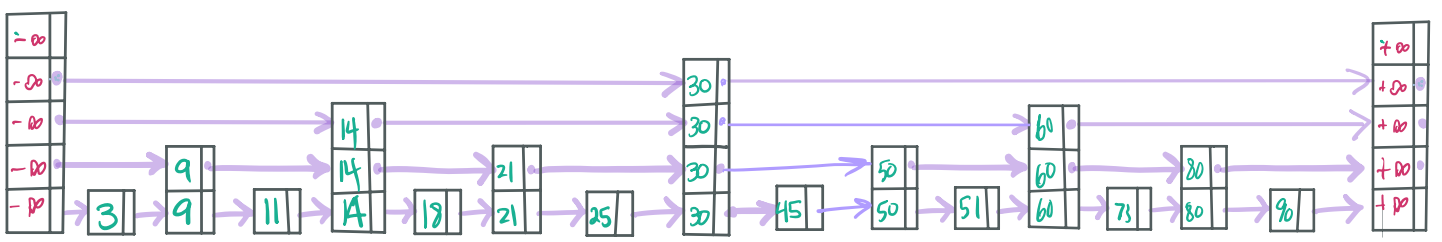
- great for insertion, deletion once you have searched and found the right spot
- not so great for search.

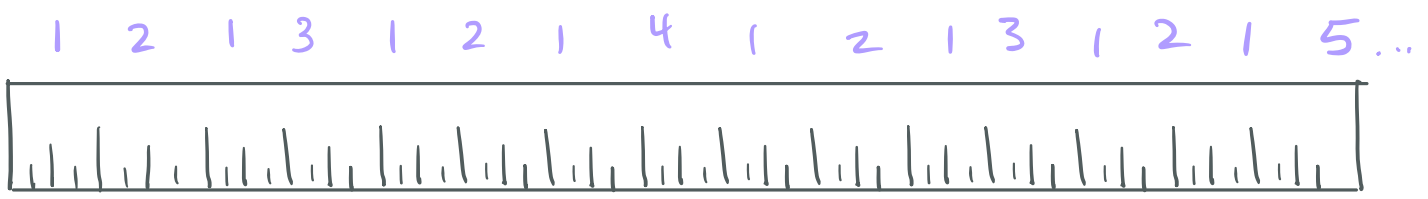
Best of Both Worlds:

have a linked list with pointers to halfway, $\frac{1}{4}$-way, $\frac{1}{8}$-way, ... $\frac{1}{2^n}$ way through list.



search(18)

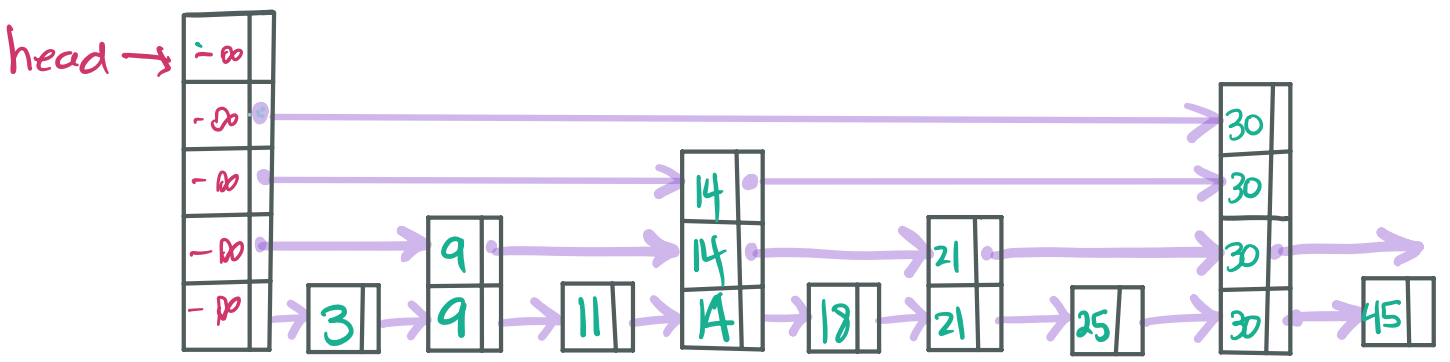






bit that flips

1
2
1
3
1
2
1
4
1
2

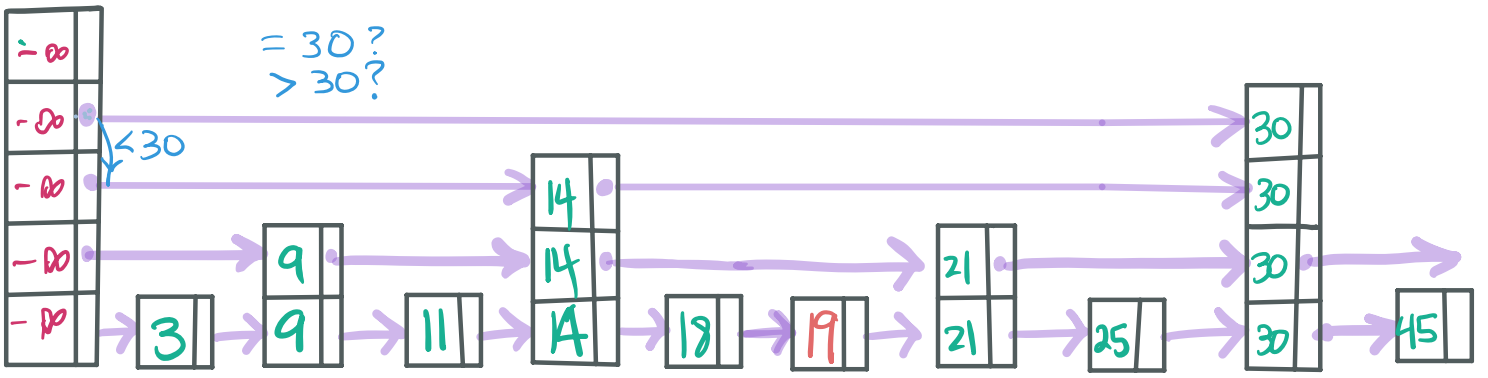


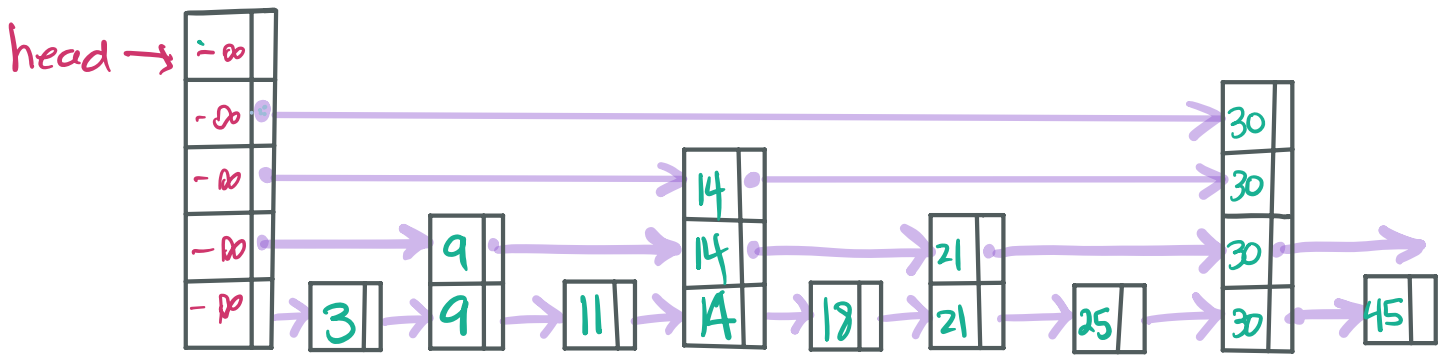
How would you search?

- like binary search

Search in Perfect Skip List:

Search (11)





How would you search?

- like binary search

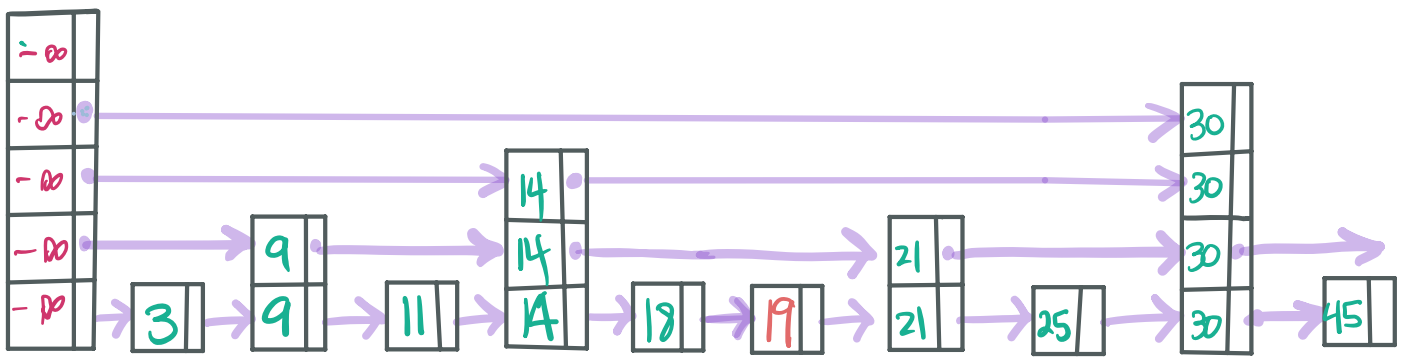
$O(\lg n)$

How would you insert?

- search for correct spot
- insert

$O(\lg n)$

But now it is not perfect.

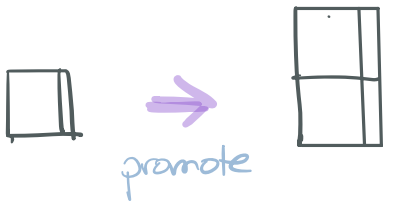


Do we keep it perfect? (1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, ...)

No. Too much work.

Instead, when a new node is inserted, we randomly select its height.

"promote" = add a layer



probability of promotion: let's say 0.5

↳ flip a coin
"heads" = promote
"tails" = no further promot

Keep flipping while see "heads"
– once you see tails, done.

$P(\text{height} = 1) = \frac{1}{2} = \frac{1}{2^1}$

$P(\text{height} = 2) = \frac{1}{4} = \frac{1}{2^2}$

$P(\text{height} = 3) = \frac{1}{8} = \frac{1}{2^3}$

$\vdots \qquad \vdots$

$P(\text{height} = \lg n) = \frac{1}{2^{\lg n}}$

Note: Skip Lists are a Randomized Data Structure

Most data actually used by applications is not very random, so if we expect it to "distribute" itself evenly, we are likely to be disappointed and bad run times may result.

But Randomized Algorithms and/or Randomized Data Structures can help

They:
- access pseudo-random bits generated by the computer
- use these to get expected behaviour to be very fast

And likelihood of bad behaviour has nothing to do with the input data, only deperdent on "luck of the draw" (and how good your pseudorandom number generator is at looking random)

see Carl Kingsford's slides on Skip Lists.