Algorithm and Data Structure Analysis (ADSA)

Skip Lists

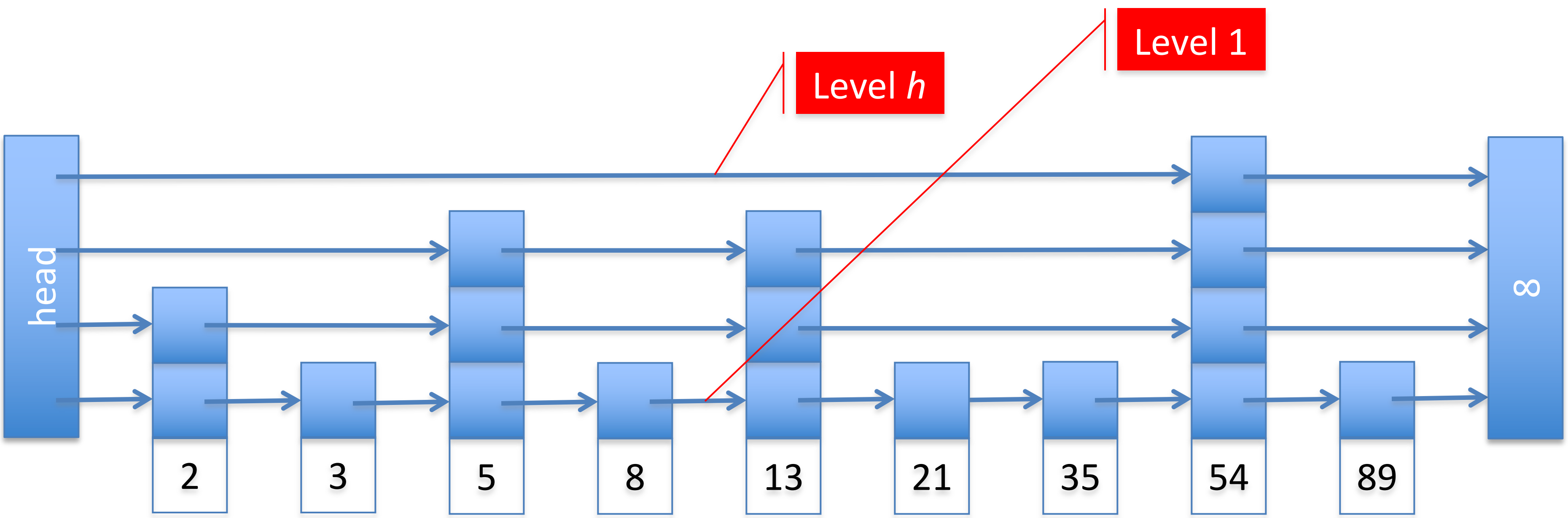
History

Invented by William Pugh (1990)

- A probabilistic data structure likely to replace balanced trees as the implementation method for many applications.
- Algorithms have the same asymptotic expected time bounds as balanced trees and are simpler.

Skip Lists

Sorted list of items, using a hierarchy of linked lists that connect increasingly sparse subsequences of the items.



Skip Lists

Theorem (without proof) Let S be a skip list containing n elements.

The expected*…

- Runtime of a search is $O(\log n)$
- Height of the skip list is $O(\log n)$
- Number of pointers is $O(2n + \lfloor \log n \rfloor + 3)$

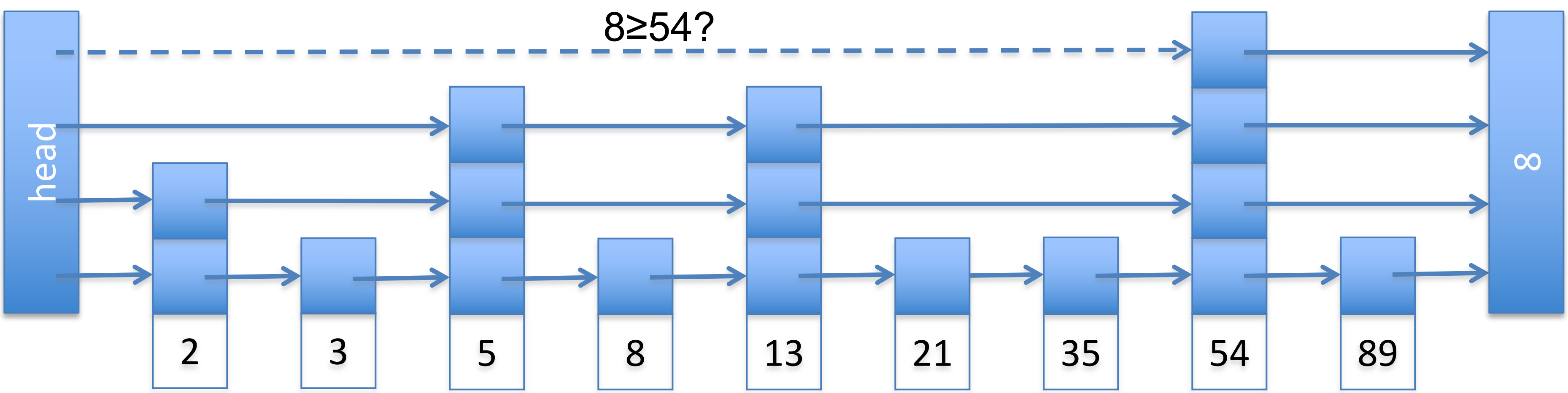
*large deviations extremely unlikely

Search(x)

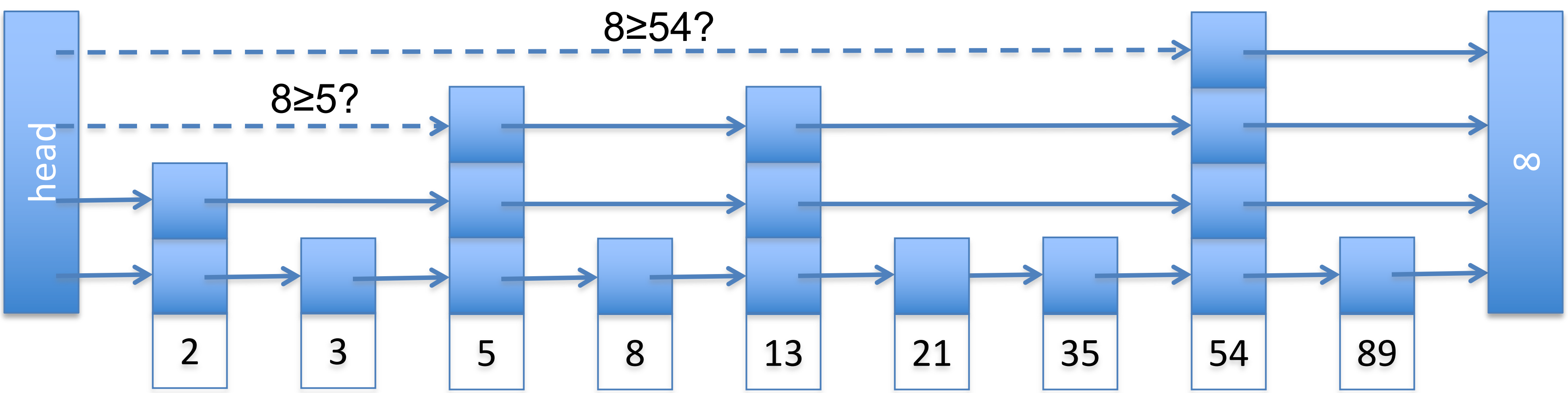
1. Start at highest level.
2. If next element $\leq x$ → go to next element
   else → descend one level

(similar to a search in a binary tree)

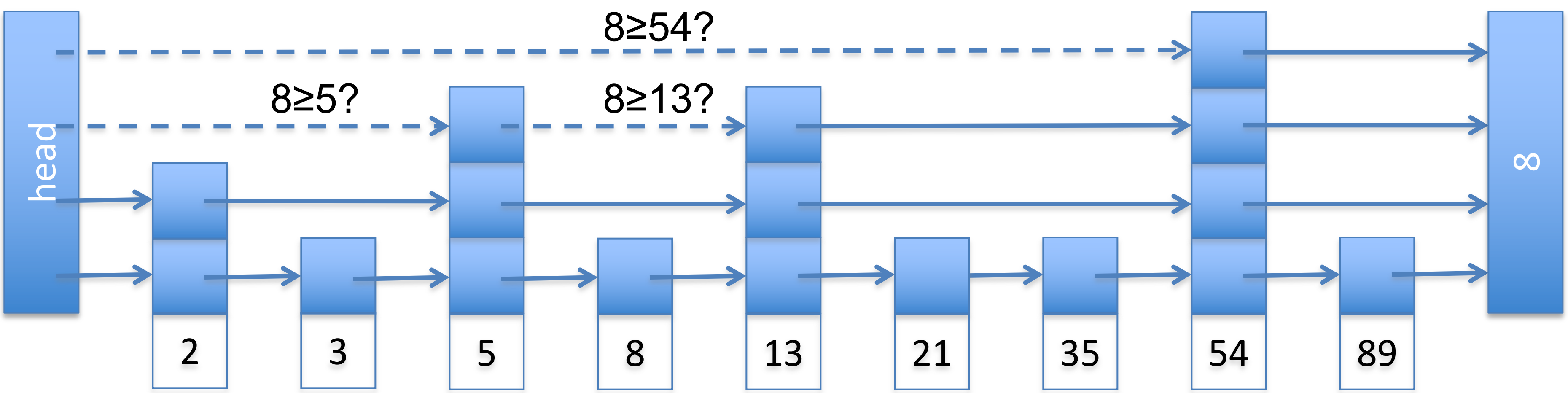
Example Search(8)



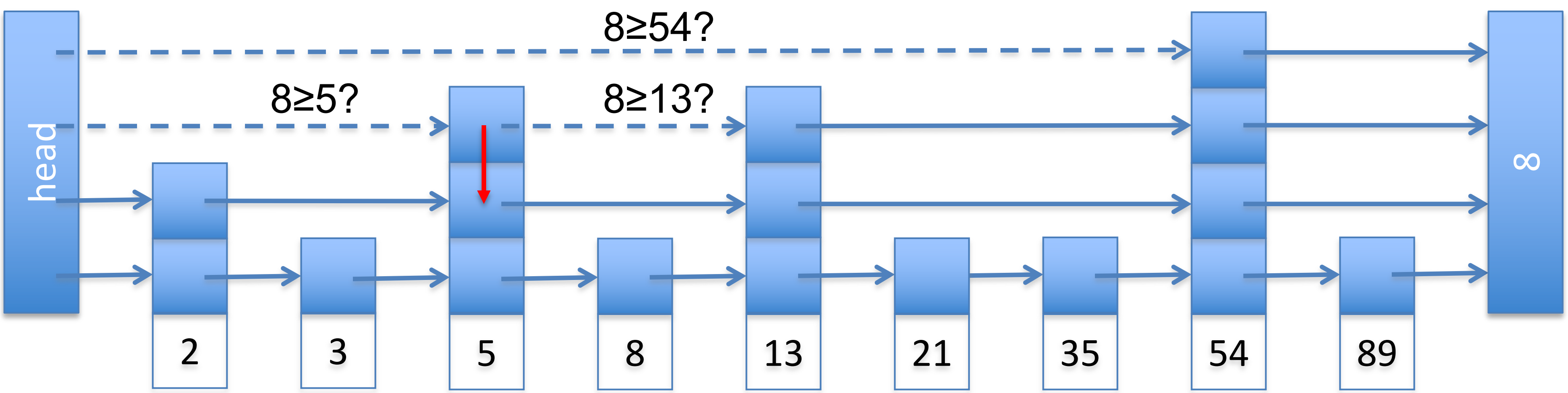
Example Search(8)



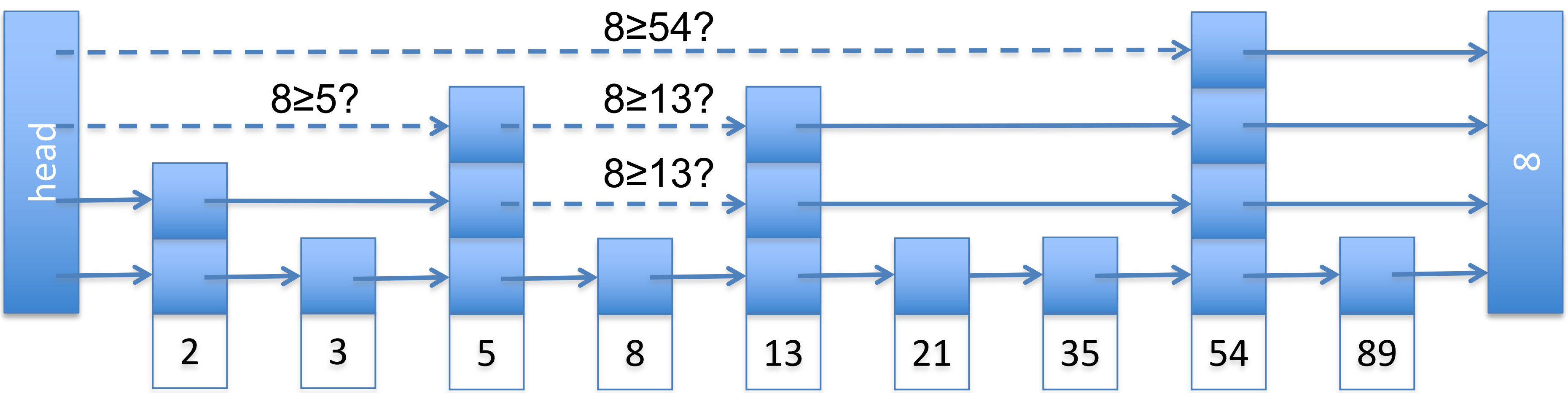
Example Search(8)



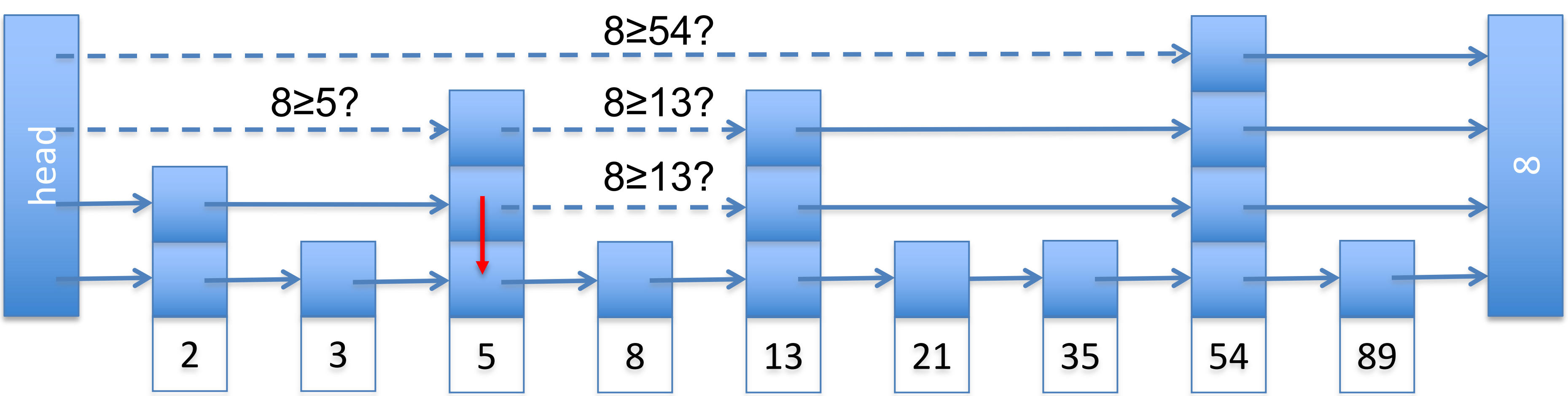
Example Search(8)



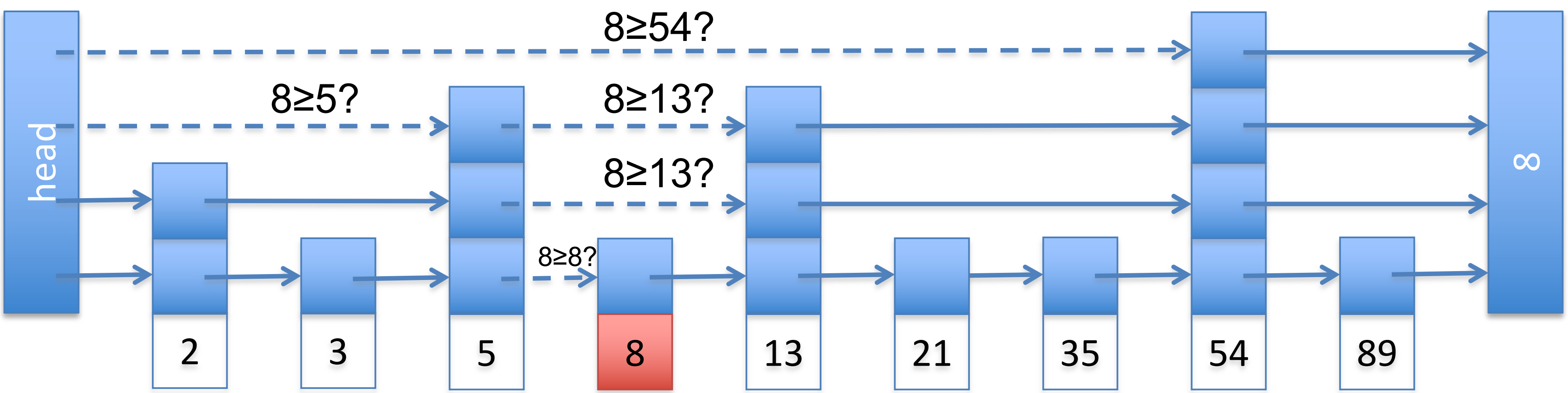
Example Search(8)



Example Search(8)



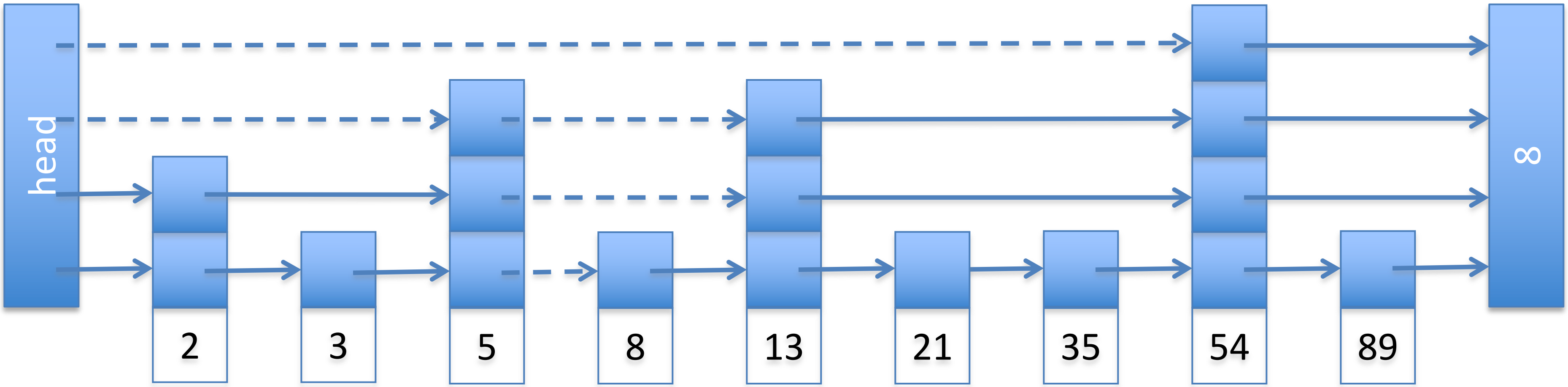
Example Search(8)



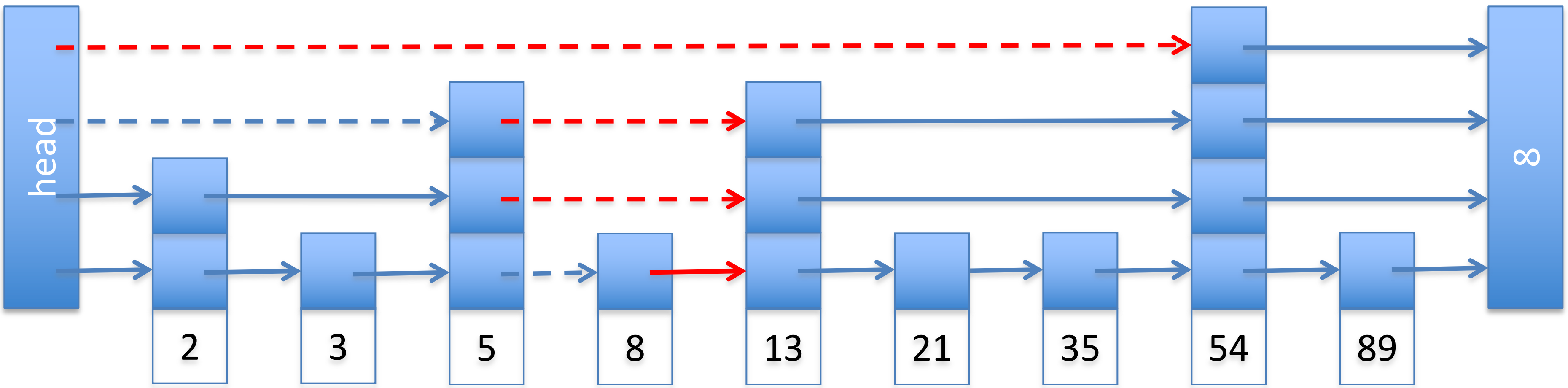
Split(x)

1. Search x.
2. Update pointers:
   - Pointers S_1 from x → point to ∞
   - Pointers S_2 "over" x → point to ∞
   - Introduce new head for 2nd list, and have it point to where S_1 and S_2 pointed to.

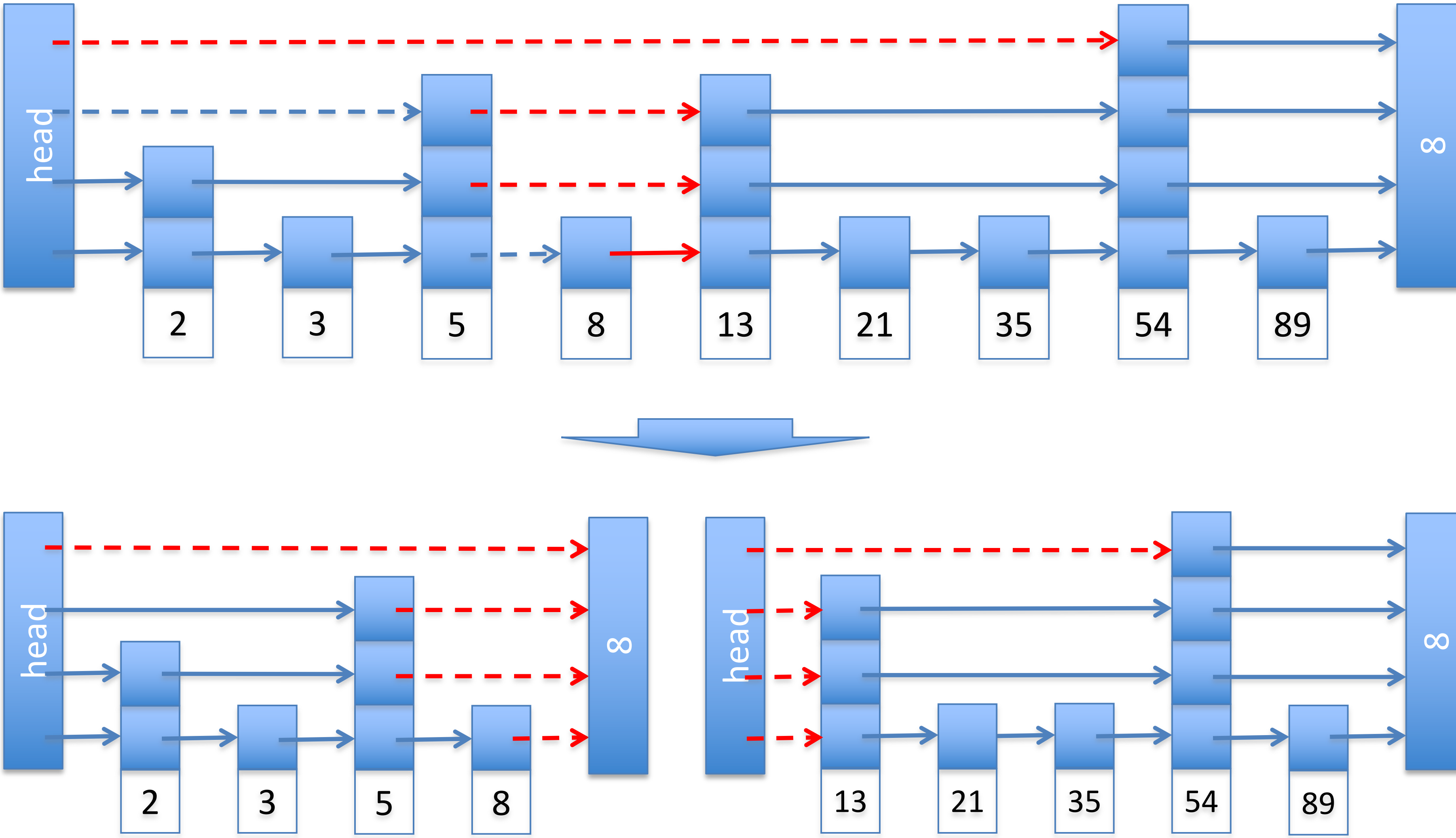
Example Split(8)



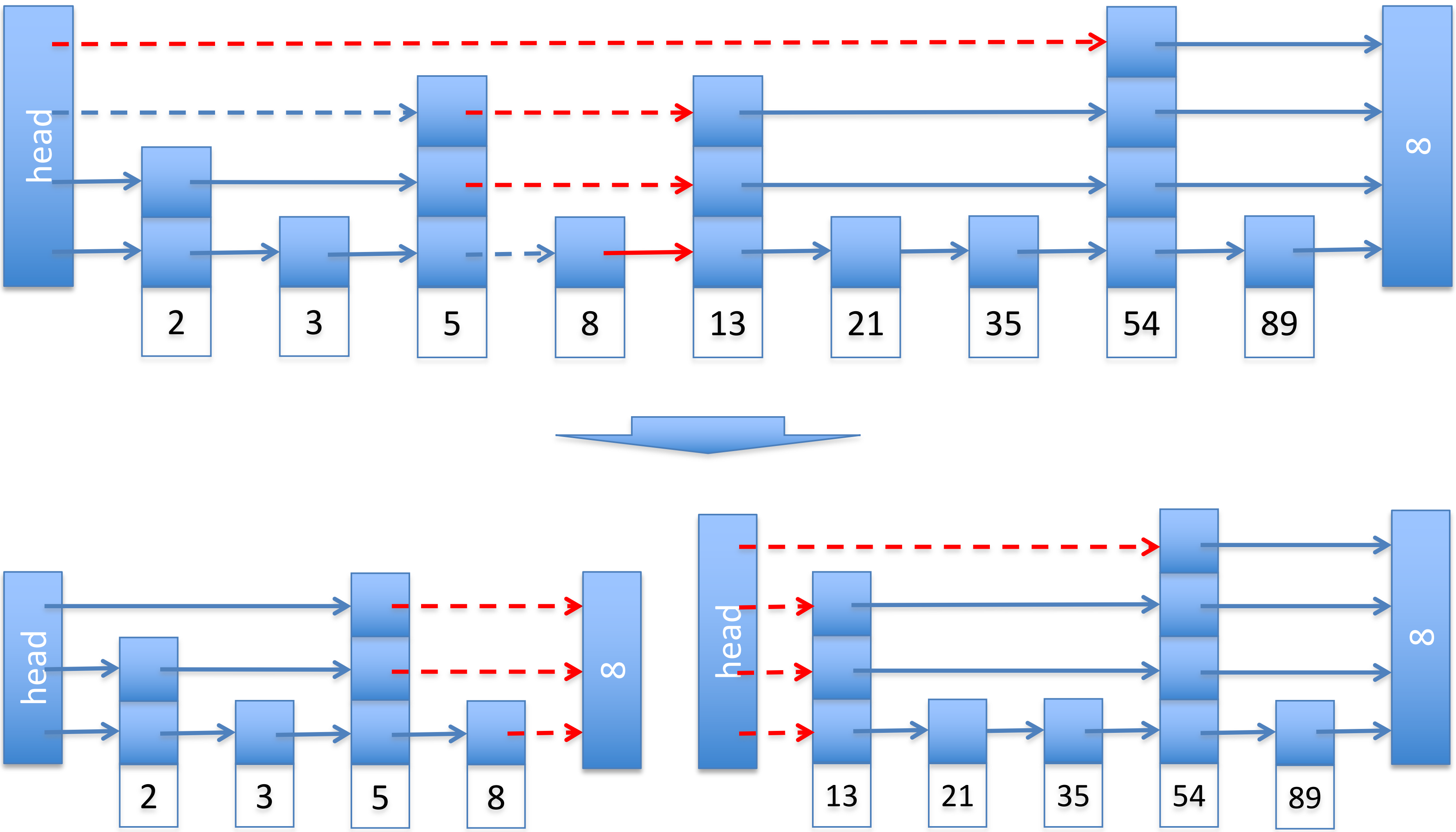
Example Split(8)



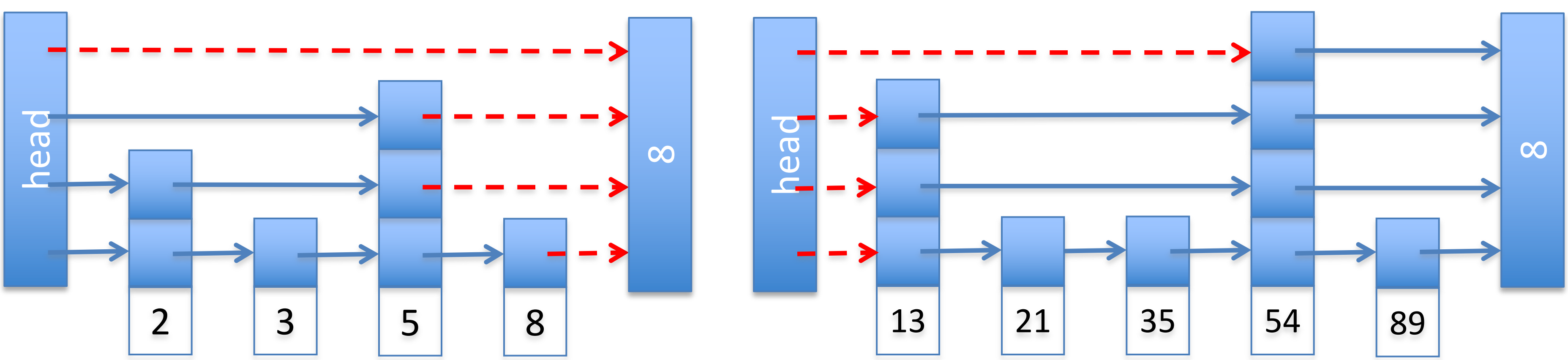
Example Split(8)



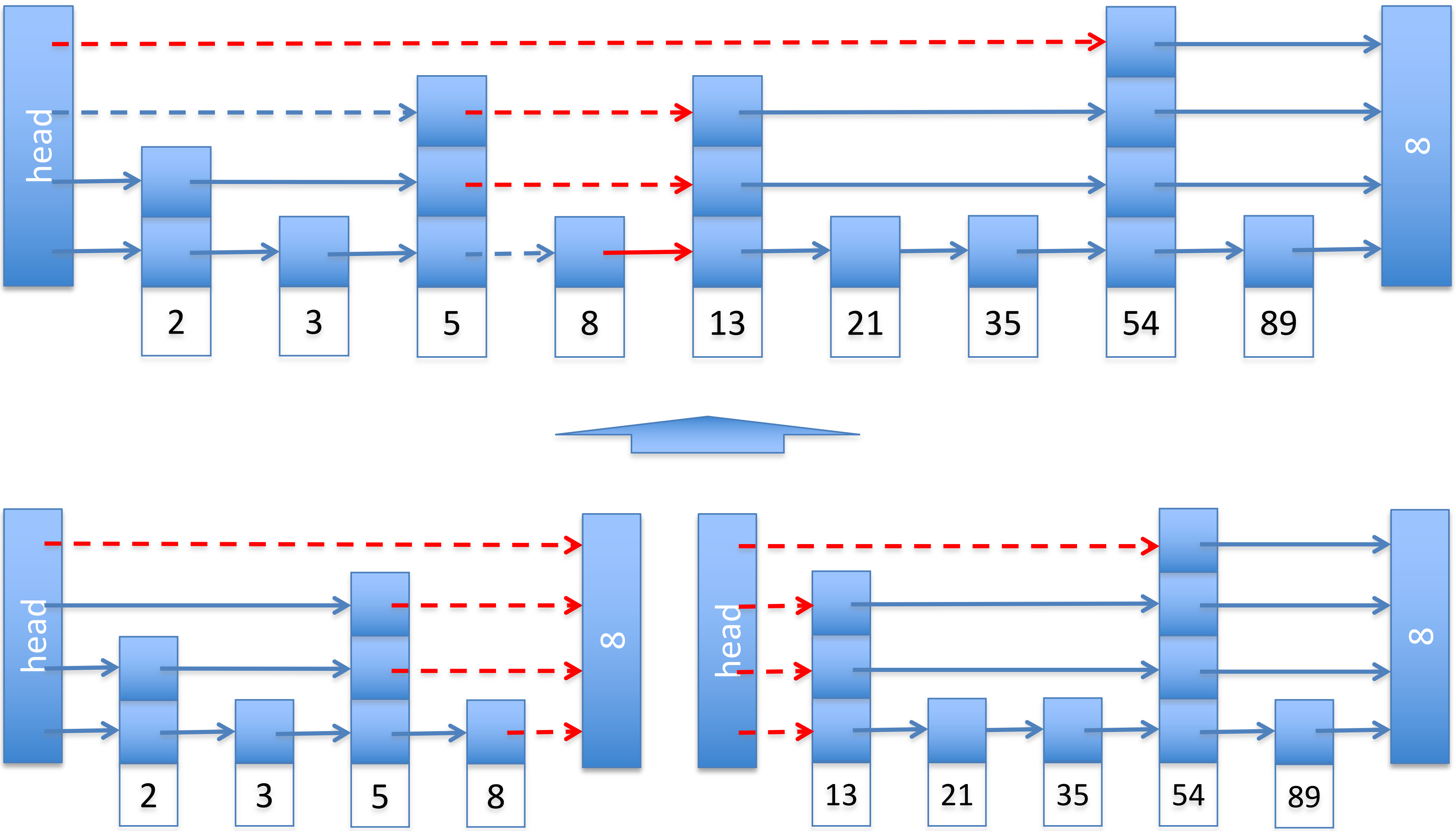
Example Split(8)



Example Concatenate



Example Concatenate

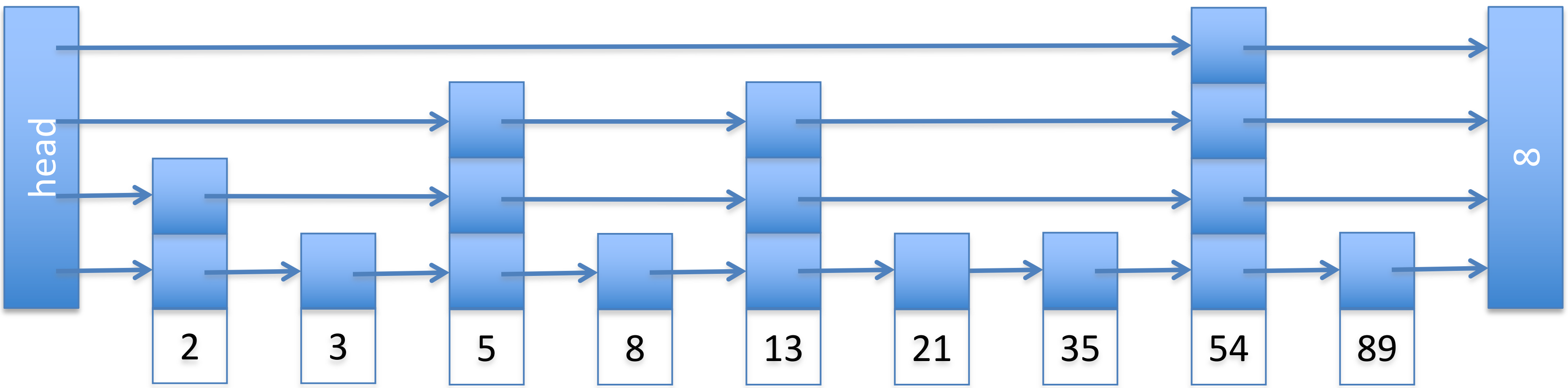


Delete(x)

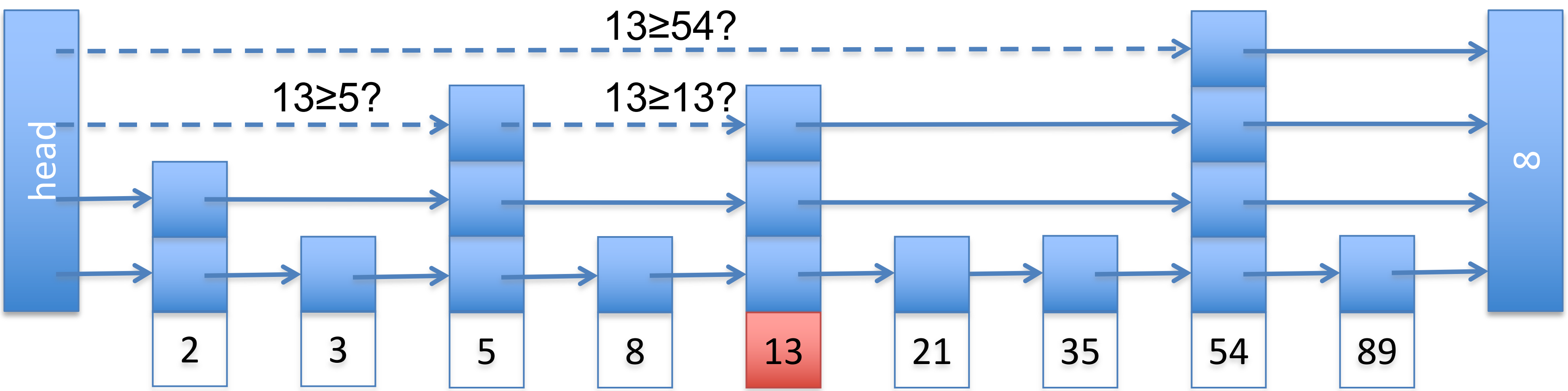
1. Search x.
2. Update pointers:

   Pointers S through x → point to where x pointed to at the corresponding levels.

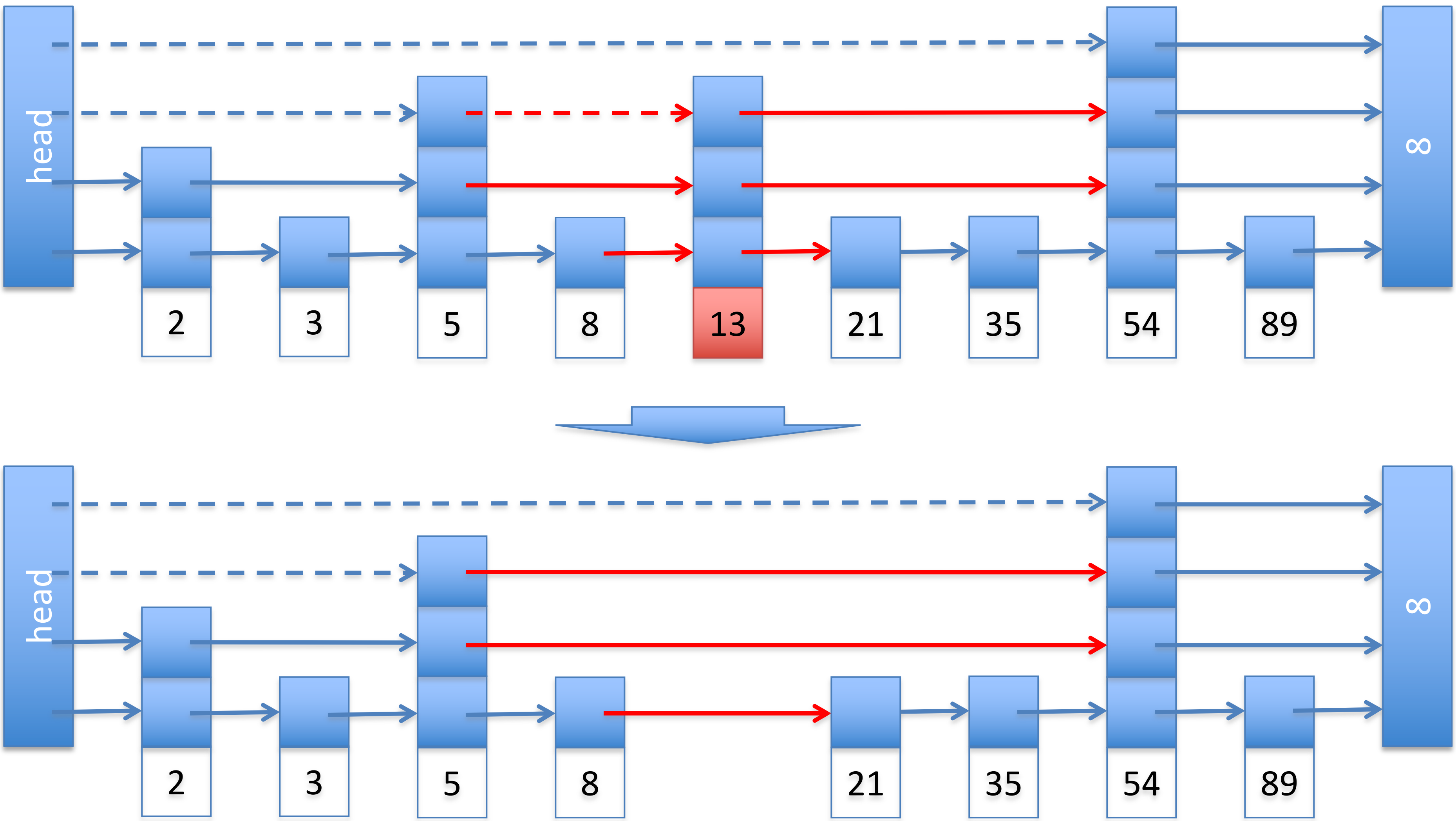
Example Delete(13)



Example Delete(13)



Example Delete(13)



Insert(x)

1. Search for x.
2. Flip coins to set the height h.

Insert(x)

1. Search for x.
2. Flip coins to set the height h.

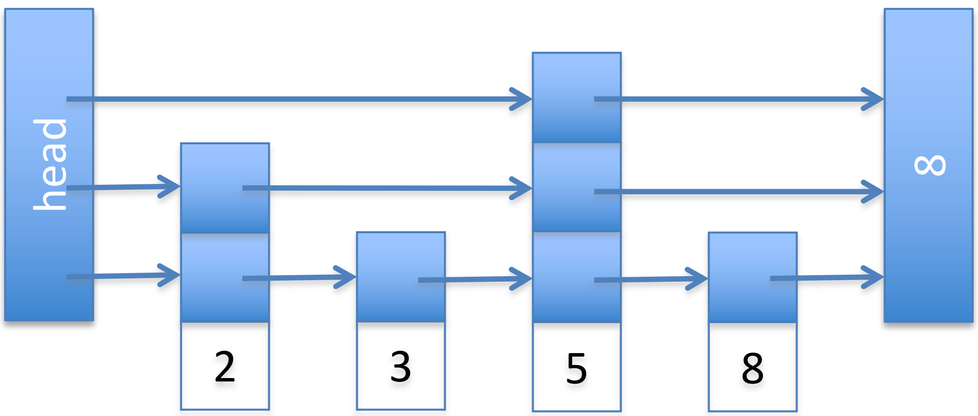
Flip a coin until "head". If h trials are needed, the height of x is h. Thus, the probability for height h is $(1/2)^h$ and the expected height is

$$\sum_{1 \le h \le \infty} h \cdot (1/2)^h = 2$$

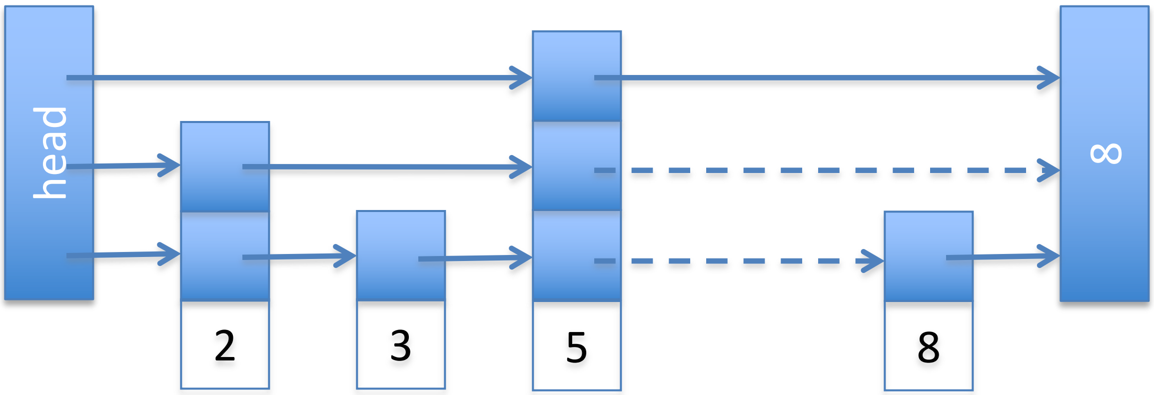
Insert(x)

1. Search for x.
2. Flip coins to set the height h.
3. Update pointers. Note that element x with h will be present in layers $1, \ldots, h$.
   Then, for each layer l:
   If $l>h$ → do nothing,
   else → we know the pointers "going through x" and update those to point to x and from x to the subsequent element.

Example Insert(7)

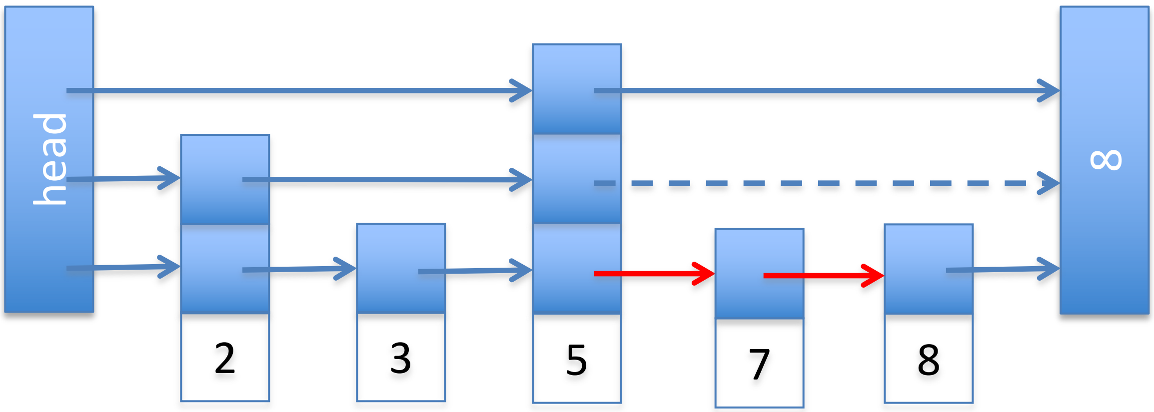


Example Insert(7)



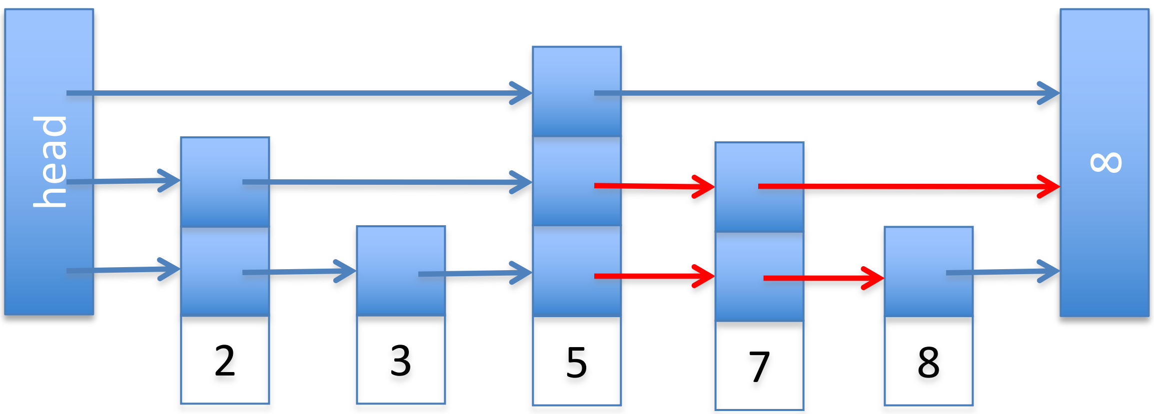
Example Insert(7)

h=2 (2 trials to get a 'head')



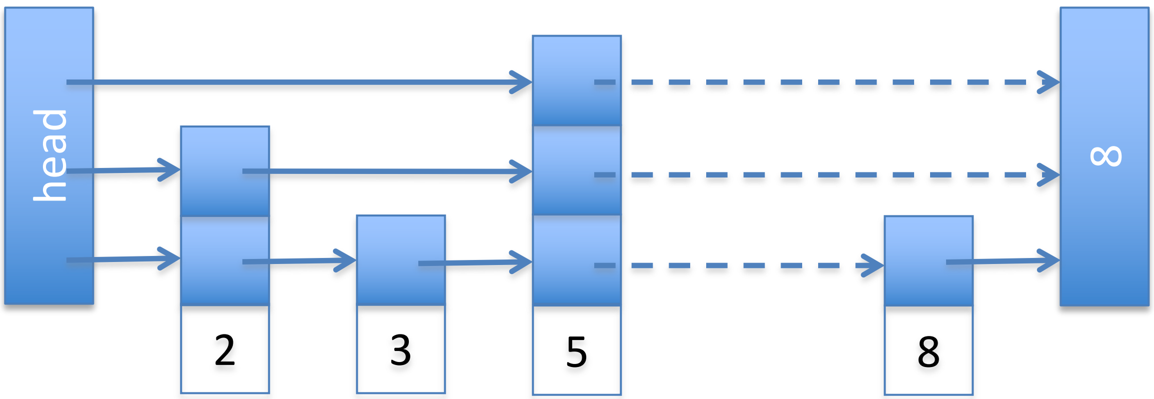
Example Insert(7)

h=2



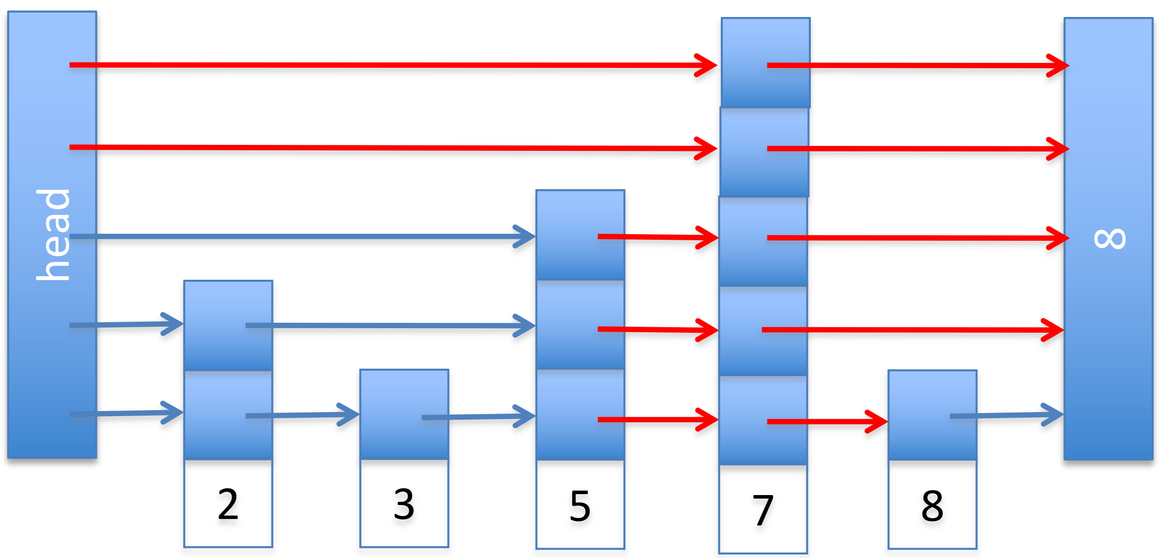
Example Insert(7)

h=5 (5 trials to get a 'head')



Example Insert(7)

h=5



Space Usage

- The space used by a skip list depends on the random bits used by each invocation of the insertion algorithm
- We use the following two basic probabilistic facts:

Fact 1: The probability of getting i consecutive heads when flipping a coin is $1/2^i$

Fact 2: If each of n entries is present in a set with probability p, the expected size of the set is np

- Consider a skip list with n entries
  - By Fact 1, we insert an entry in list S_i with probability $1/2^i$
  - By Fact 2, the expected size of list S_i is $n/2^i$
- The expected number of nodes used by the skip list is

$$\sum_{i=0}^{h} \frac{n}{2^i} = n \sum_{i=0}^{h} \frac{1}{2^i} < 2n$$

- Thus, the expected space usage of a skip list with n items is $O(n)$

Height

- The running time of the search is affected by the height h of the skip lists
- We show that with high probability, a skip list with n items has height $O(\log n)$
- We use the following additional probabilistic fact:

Fact 3: If each of n events has probability p, the probability that at least one event occurs is at most np

- Consider a skip list with n entires
  - By Fact 1, we insert an entry in list S_i with probability $1/2^i$
  - By Fact 3, the probability that list S_i has at least one item is at most $n/2^i$
- By picking $i = \mathrm{c}\log n$, we have that the probability that $S_{\mathrm{c}\log n}$ has at least one entry is at most
  $n/2^{\mathrm{c}\log n} = n/n^{\mathrm{c}} = 1/n^{\mathrm{c}-1}$
- Thus a skip list with n entries has height at most $O(\log n)$ with probability at least $1 - 1/n^{\mathrm{c}-1}$ that is asymptotically 1 for large constant c.