CS 261: Data Structures

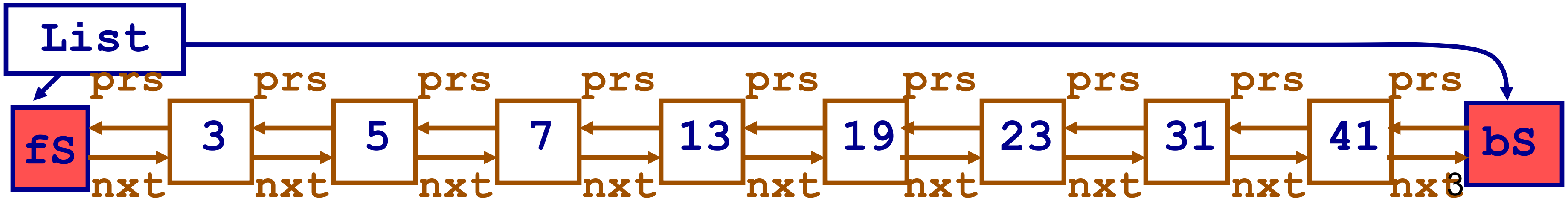
Skip Lists

Complexity – Lists and Arrays

OPERATIONS	ORDINARY LISTS AND ARRAYS	SORTED ARRAYS	SORTED LISTS
Add	O(1)	O(n)	O(n)
Remove	O(n)	O(n)	O(n)
Contains	O(n)	O(log n)	O(n)

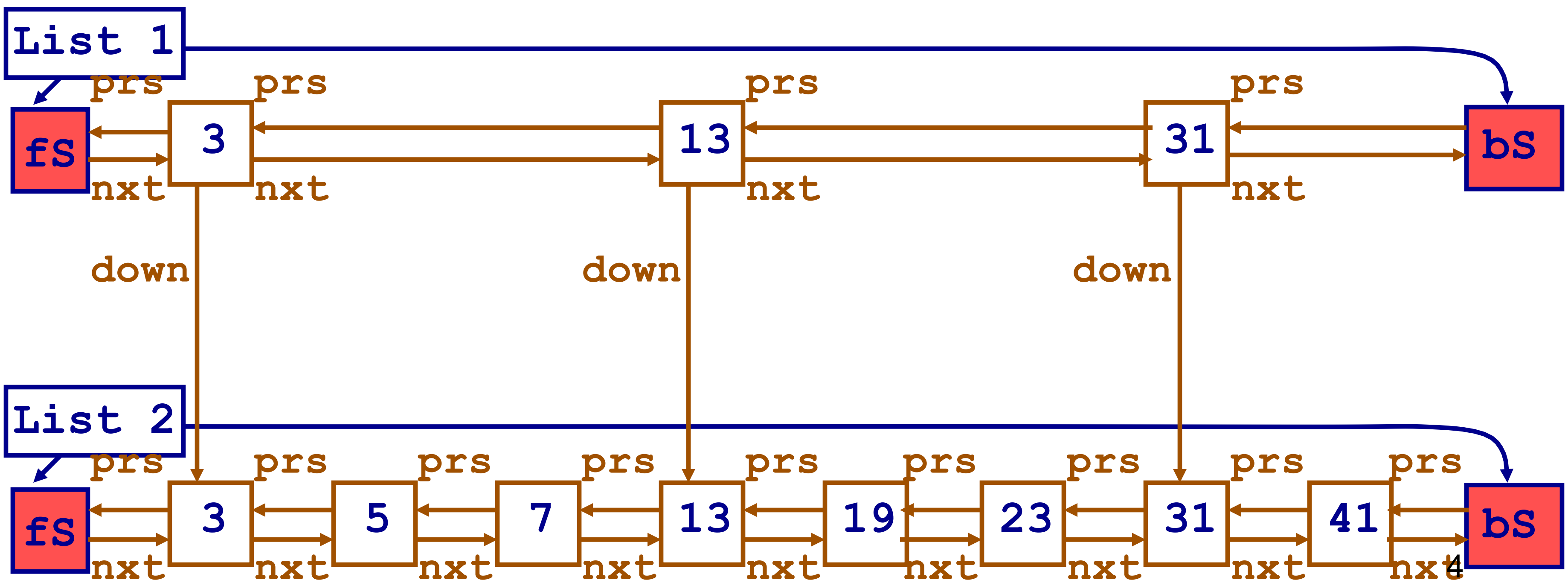
Sorted Linked Lists

- How to improve complexity of a sorted linked list?
- We could use two sorted linked lists, with pointers between them



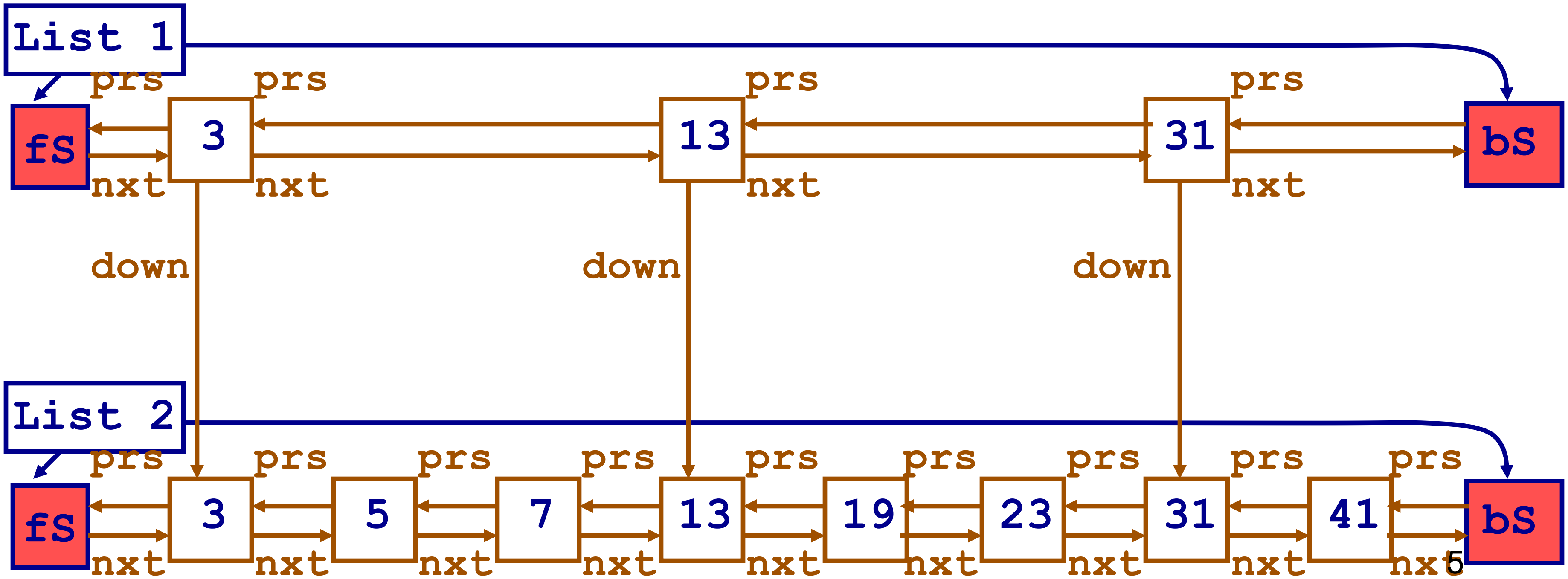
Two Sorted Linked Lists

- Constructed from the same elements
- Establish pointers between equal links



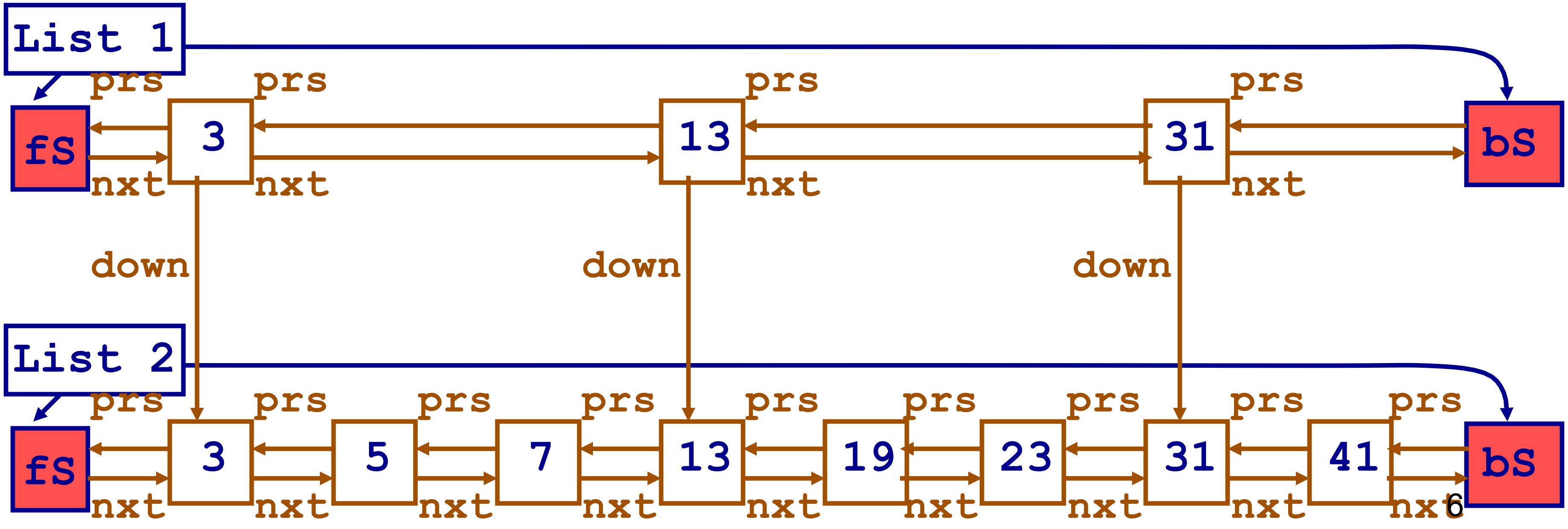
Motivation

- Regular Trains vs. Express Trains



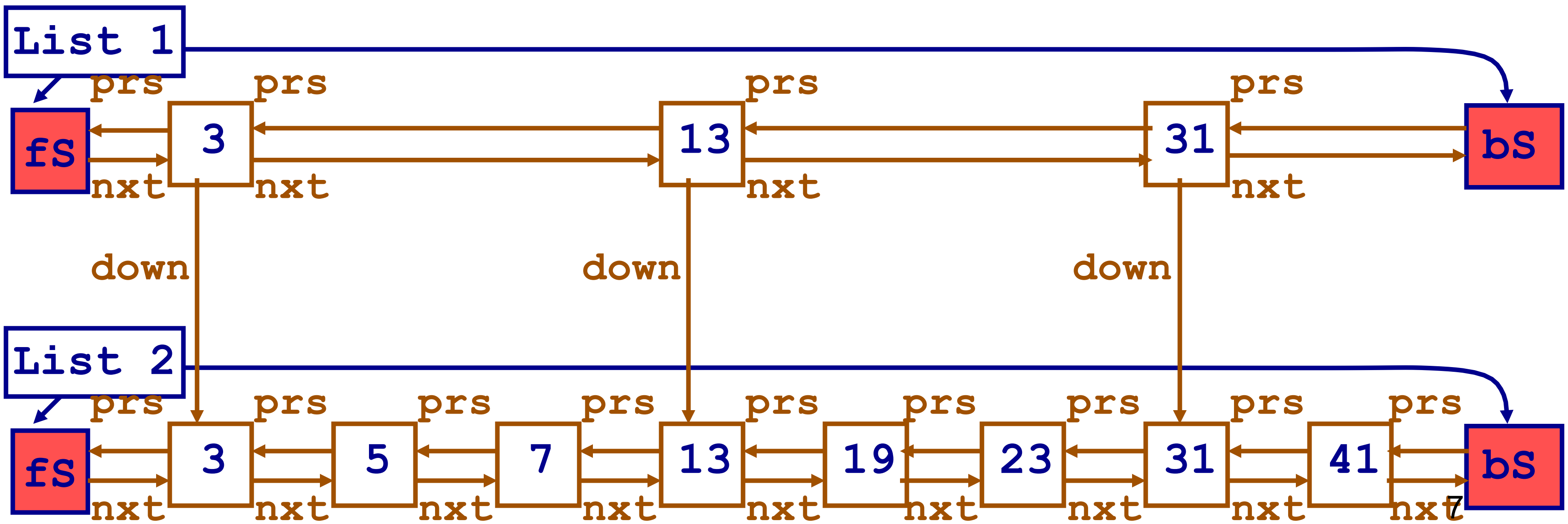
Two Sorted Linked Lists

- List 2: stores all elements
- List 1: stores only a subset of elements



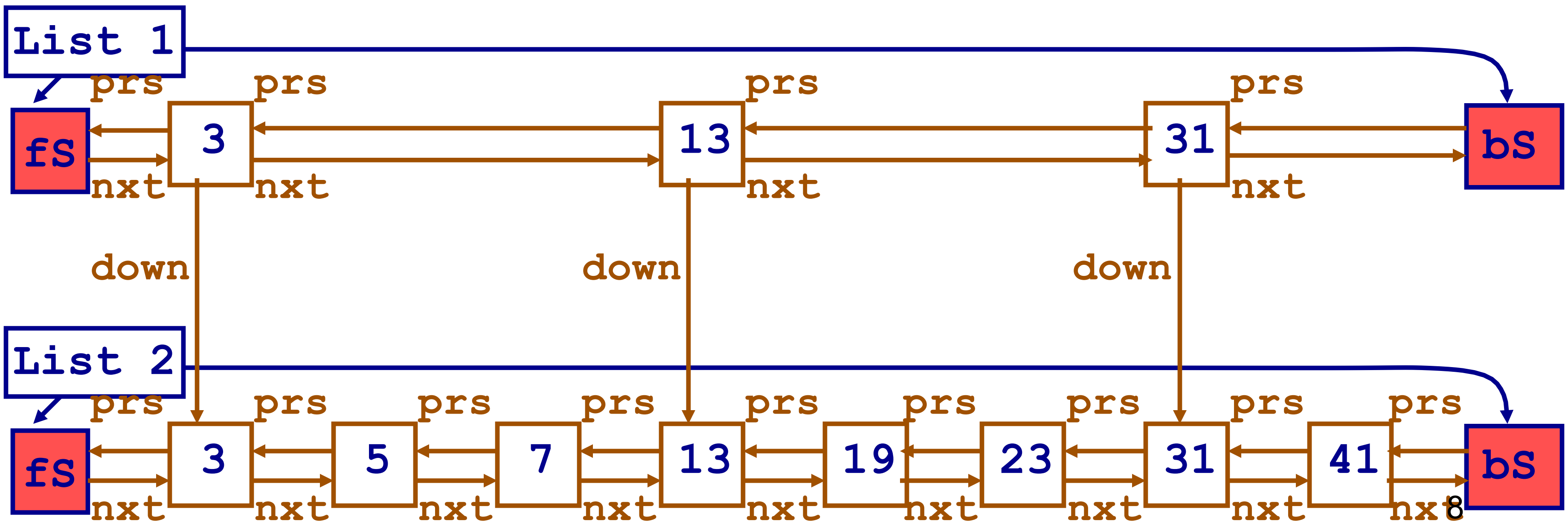
How to Search for an Element?

- We start from the 1st element of List 1
- Stay on the "express line" as long as you can
- Then, take the "local line"



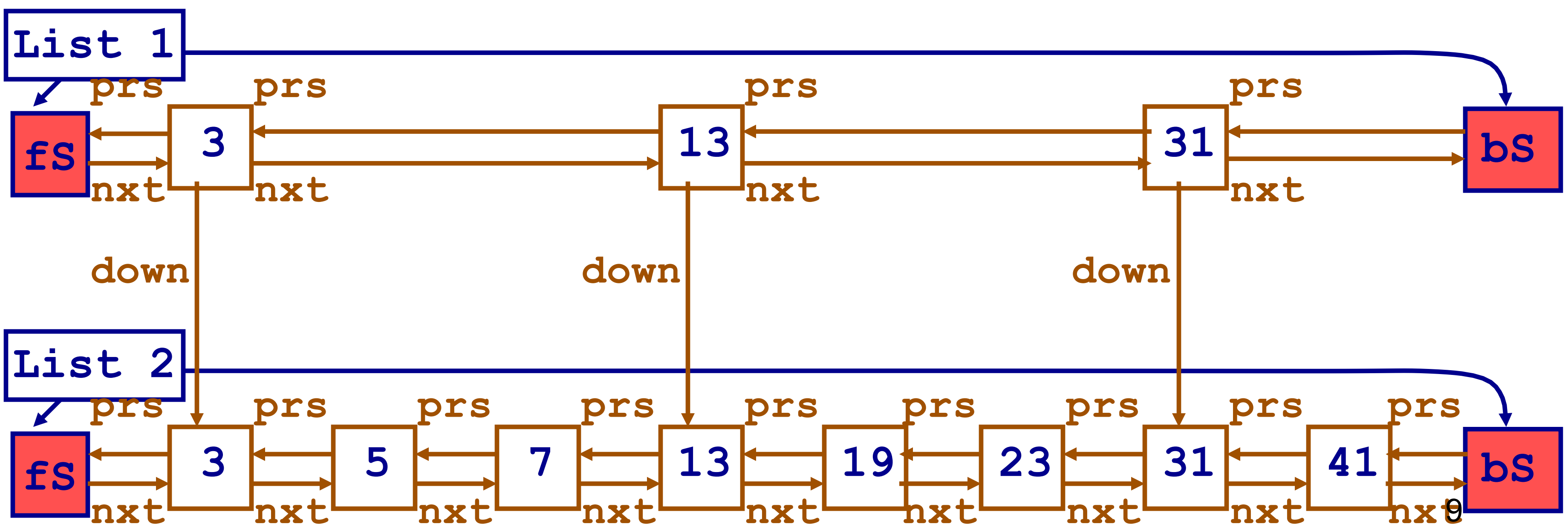
How to Choose Elements for List 1?

- Goal: Maximize fast access to all elements
- List 1 picks uniformly a subset of elements (e.g. every 2nd, or 3rd, or 4th … element)

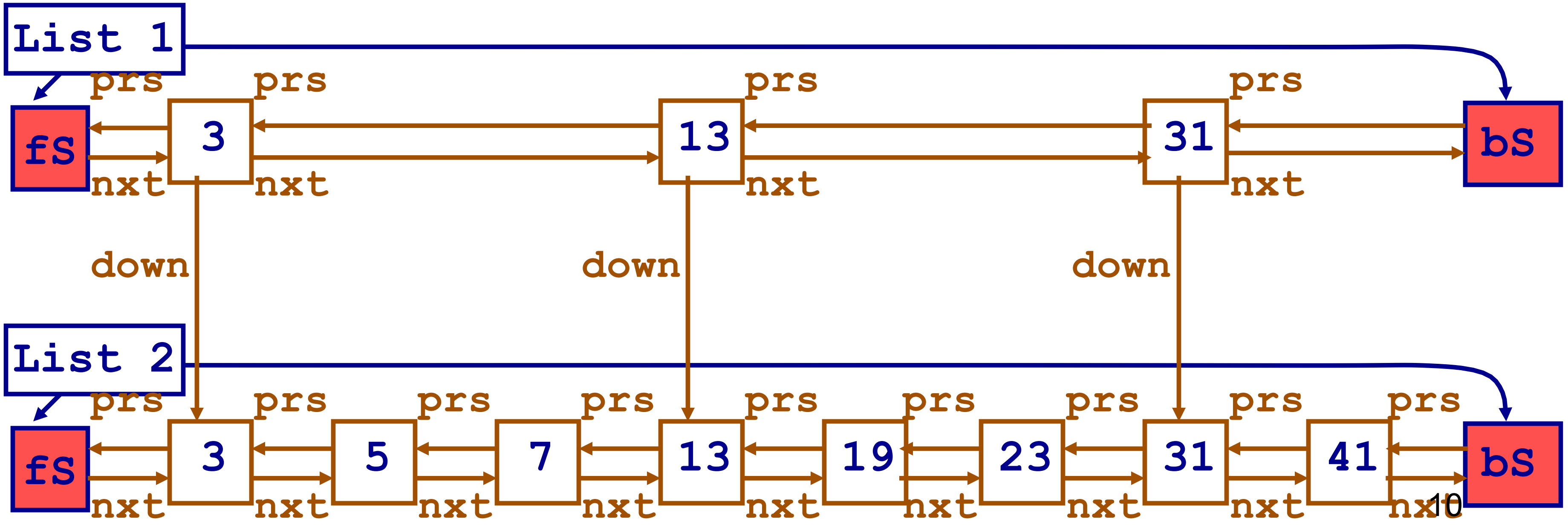


How to Choose Elements for List 1?

- But how <u>exactly</u> to uniformly sample elements for List 1?



The Goal: Minimize Complexity of Search

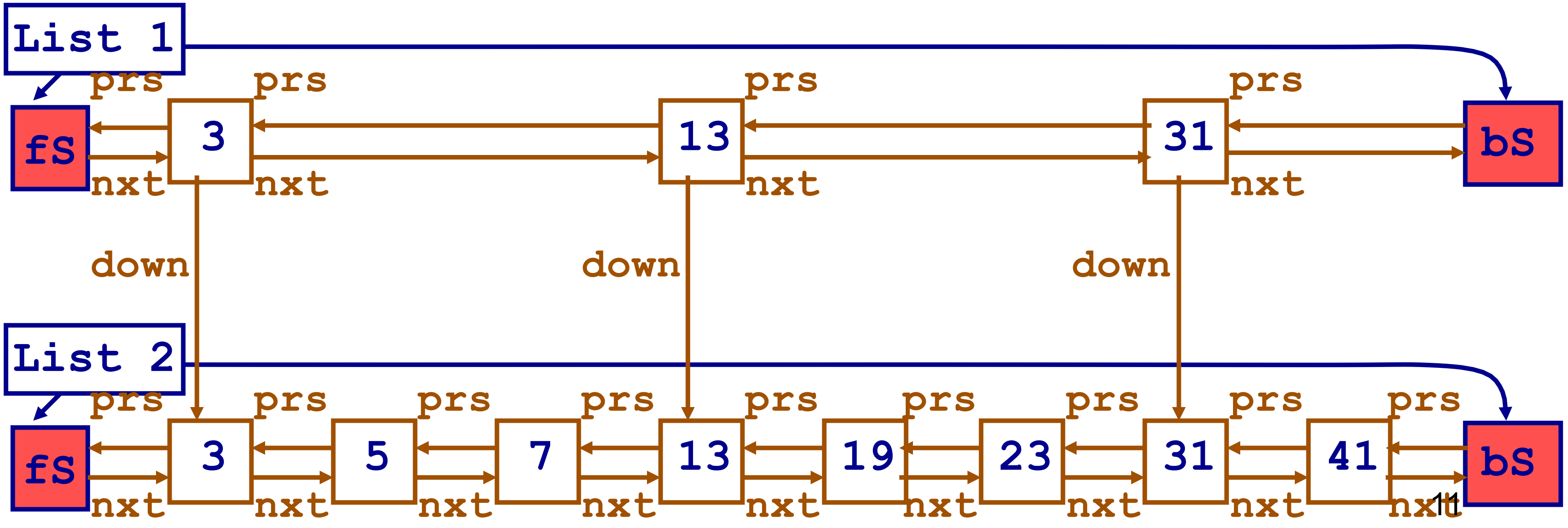


What is Complexity of Search?

num. of elements in List 1 $\longrightarrow$

$$|L_1| + \frac{|L_2|}{|L_1|}$$

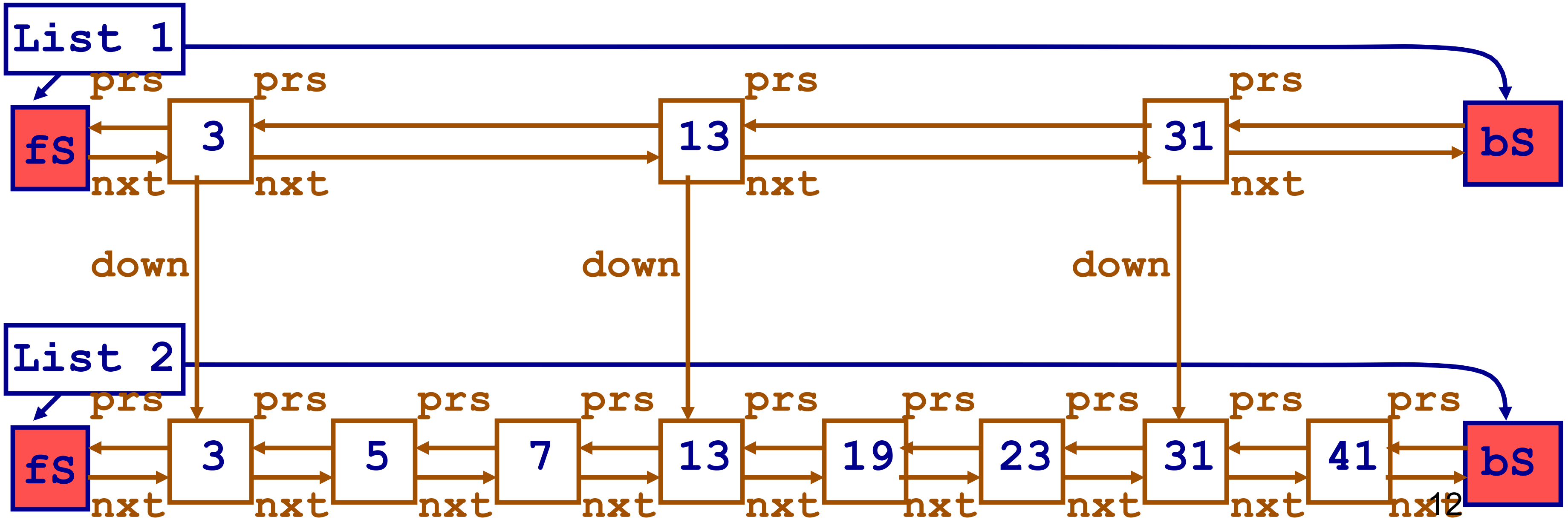
$\longleftarrow$ num. of elements in a segment of List 2



What is Complexity of Search?

$$x = \quad |L_1| + \frac{|L_2|}{|L_1|} = n \text{ input}$$

$$\frac{|L_2|}{|L_1|} = x$$



What is Complexity of Search?

minimize $$x + \frac{n}{x}$$

What is Complexity of Search?

minimum
of the function

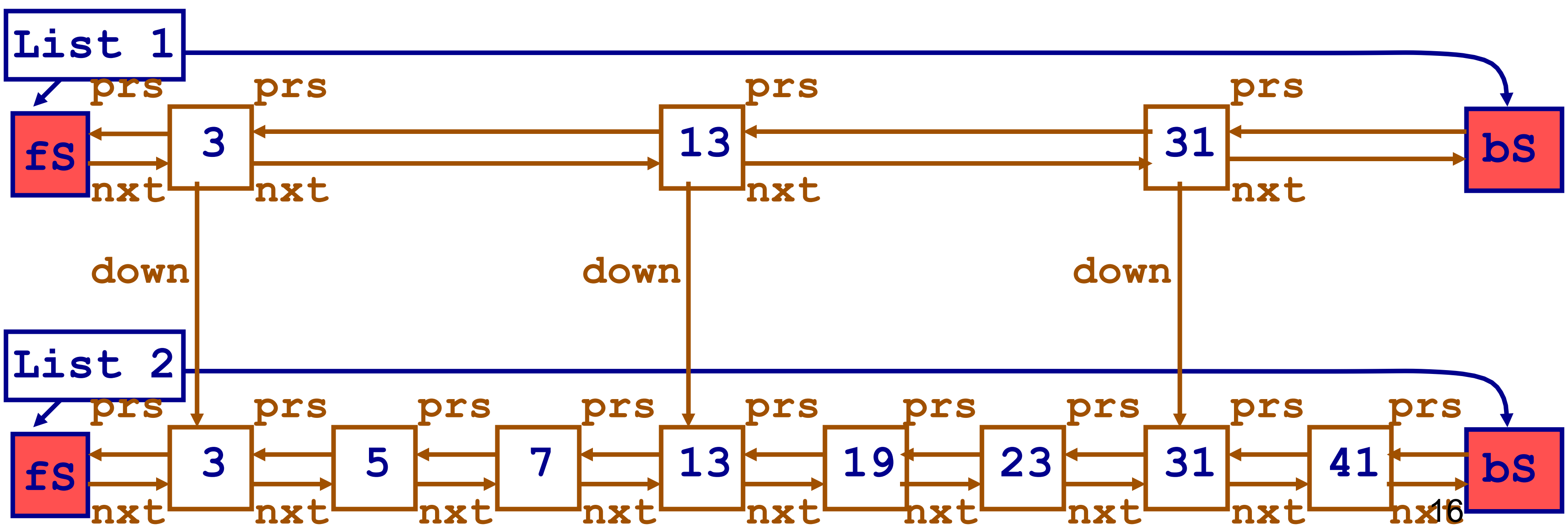
$$\frac{d\left(x + \dfrac{n}{x}\right)}{dx} = 0$$

What is Complexity of Search?

$$\frac{d\left(x + \dfrac{n}{x}\right)}{dx} = 1 - \frac{n}{x^2} = 0 \quad \Rightarrow \quad x = \sqrt{n}$$

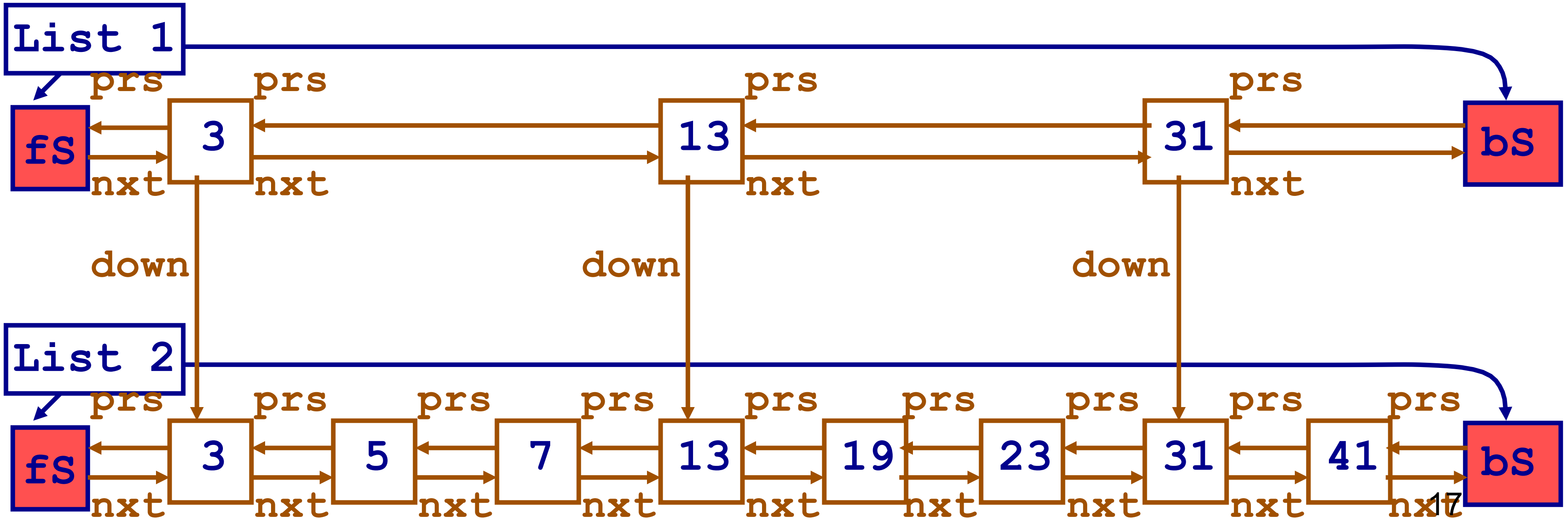
What is Complexity of Search?

$$\sqrt{n} = |L_1| + \frac{|L_2|^{\;= n}}{|L_1|_{\;= \sqrt{n}}}$$



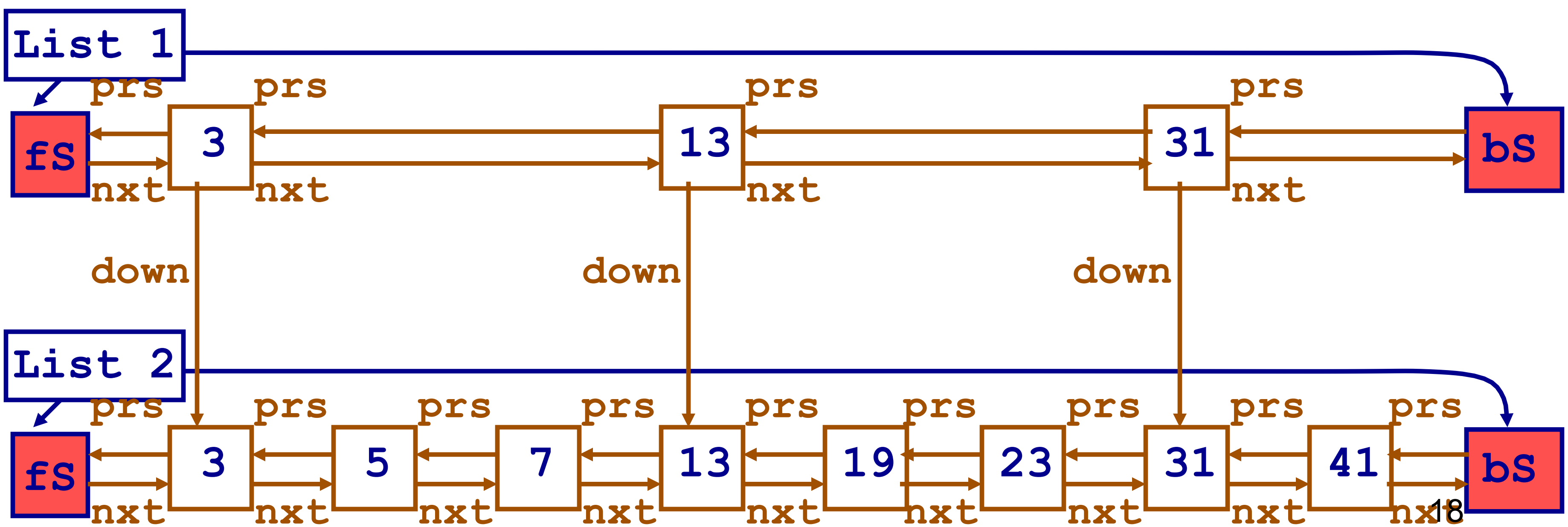
What is Complexity of Search?

$$2\sqrt{n}$$



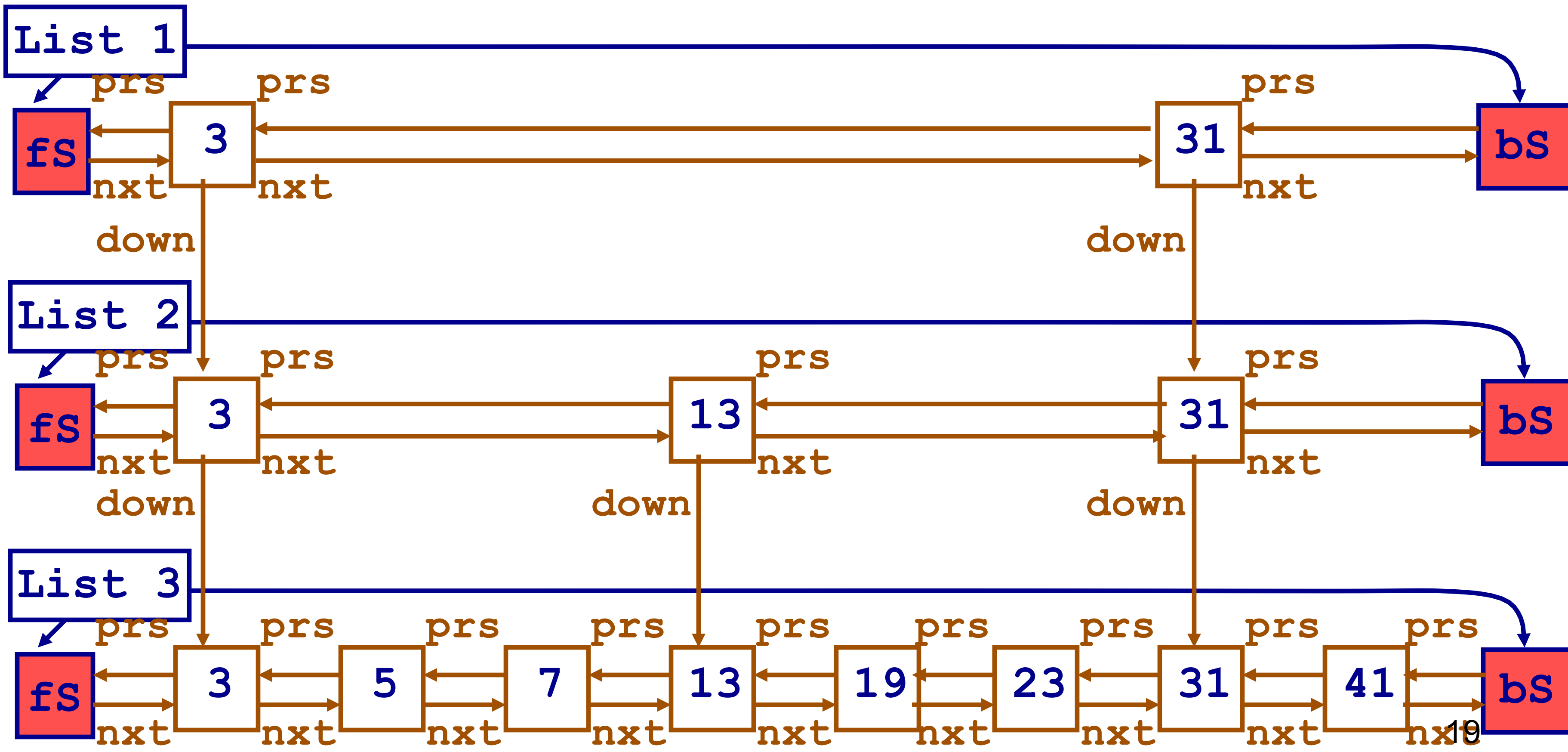
What is Complexity of Search?

two sorted lists $O(\sqrt{n}) < O(n)$ one sorted list



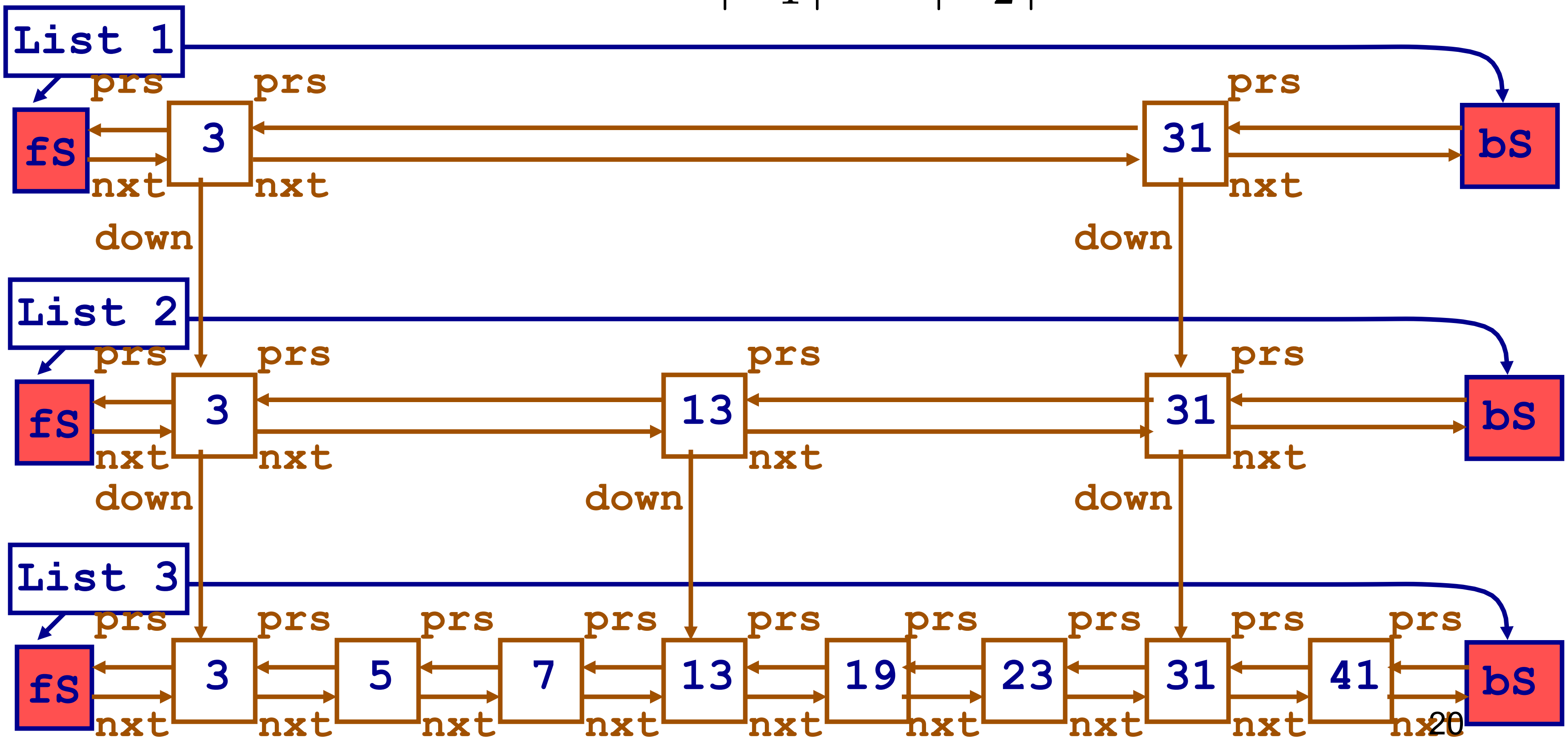
How Many Lists Should We Form?

If 2 lists have lower complexity, maybe 3 lists will have even lower complexity



What is Complexity for 3 Lists?

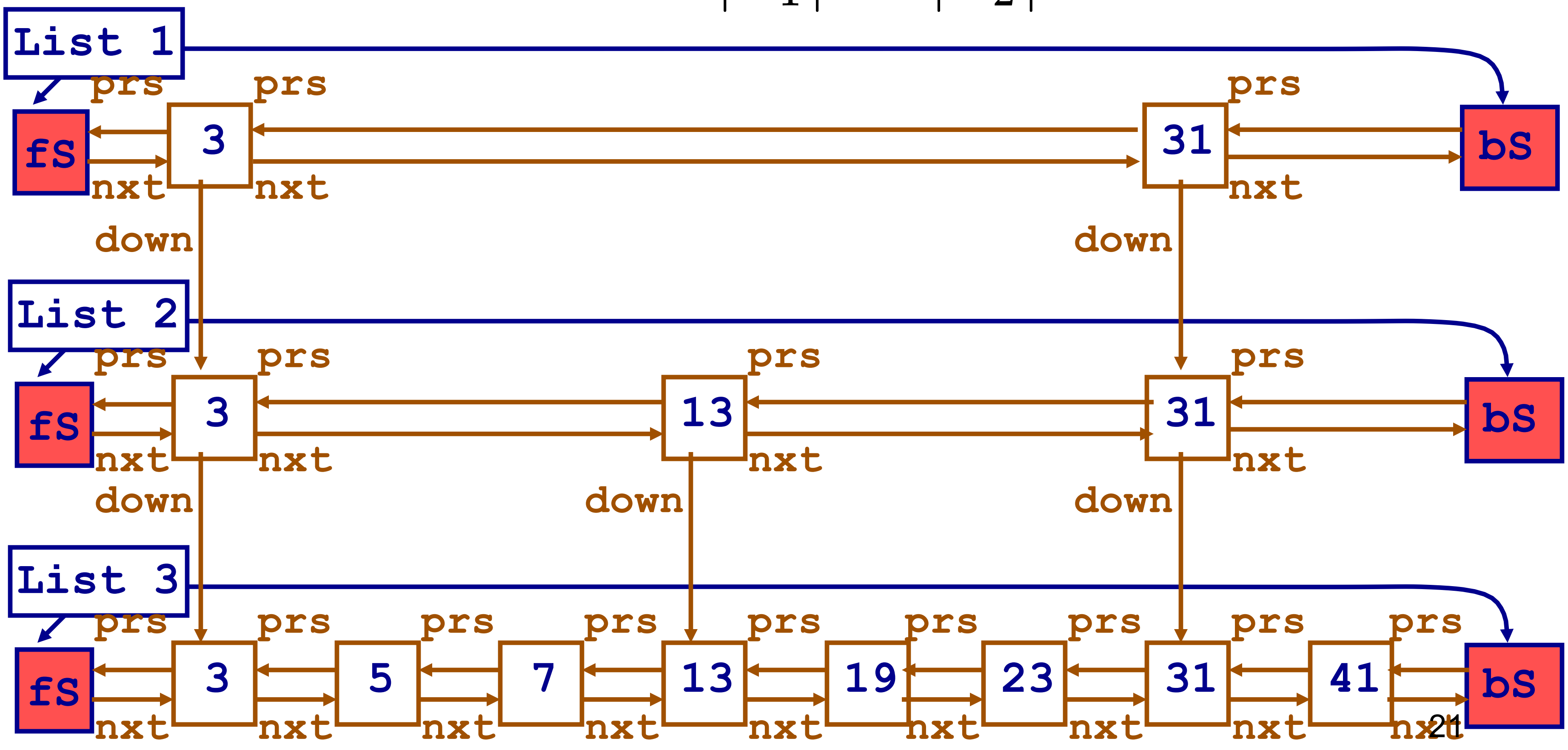
$$|L_1| + \frac{|L_2|}{|L_1|} + \frac{|L_3|}{|L_2|}$$



What is Complexity for 3 Lists?

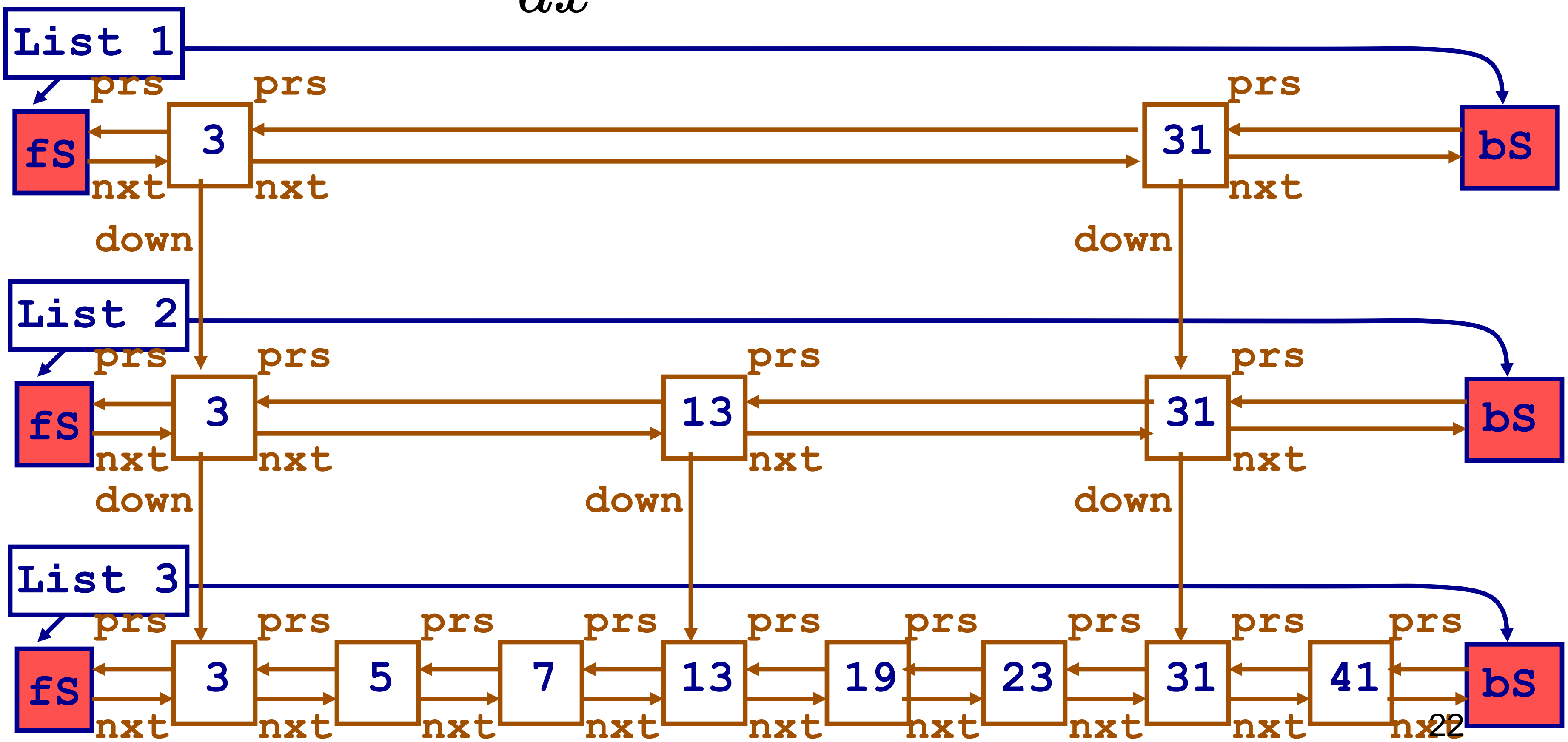
$$x = |L_1| + \frac{|L_2|}{|L_1|} + \frac{|L_3|}{|L_2|} \quad = n \text{ input}$$

$= x^2$



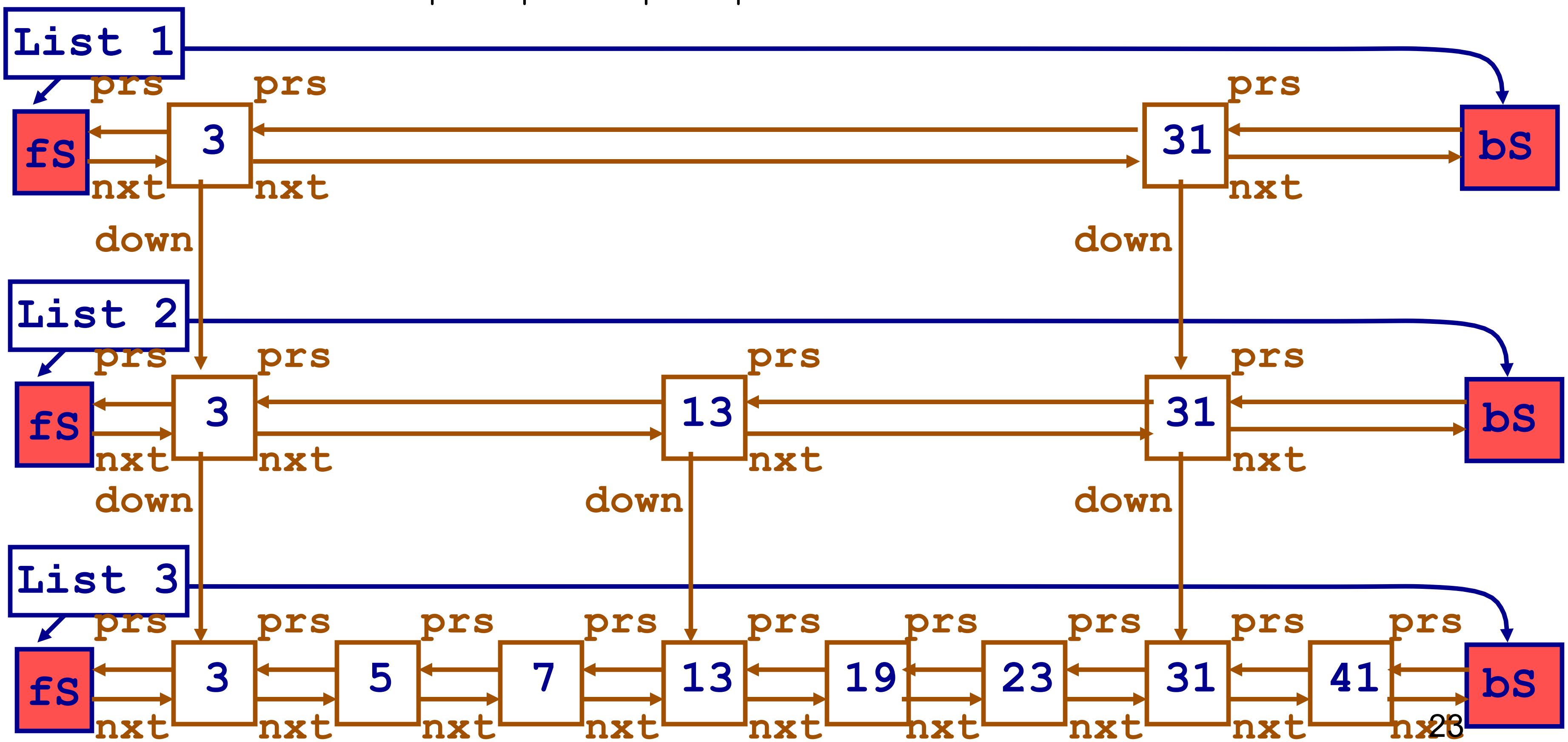
What is Complexity for 3 Lists?

$$\frac{d(x + x + n/x^2)}{dx} = 0 \Rightarrow x = \sqrt[3]{n}$$



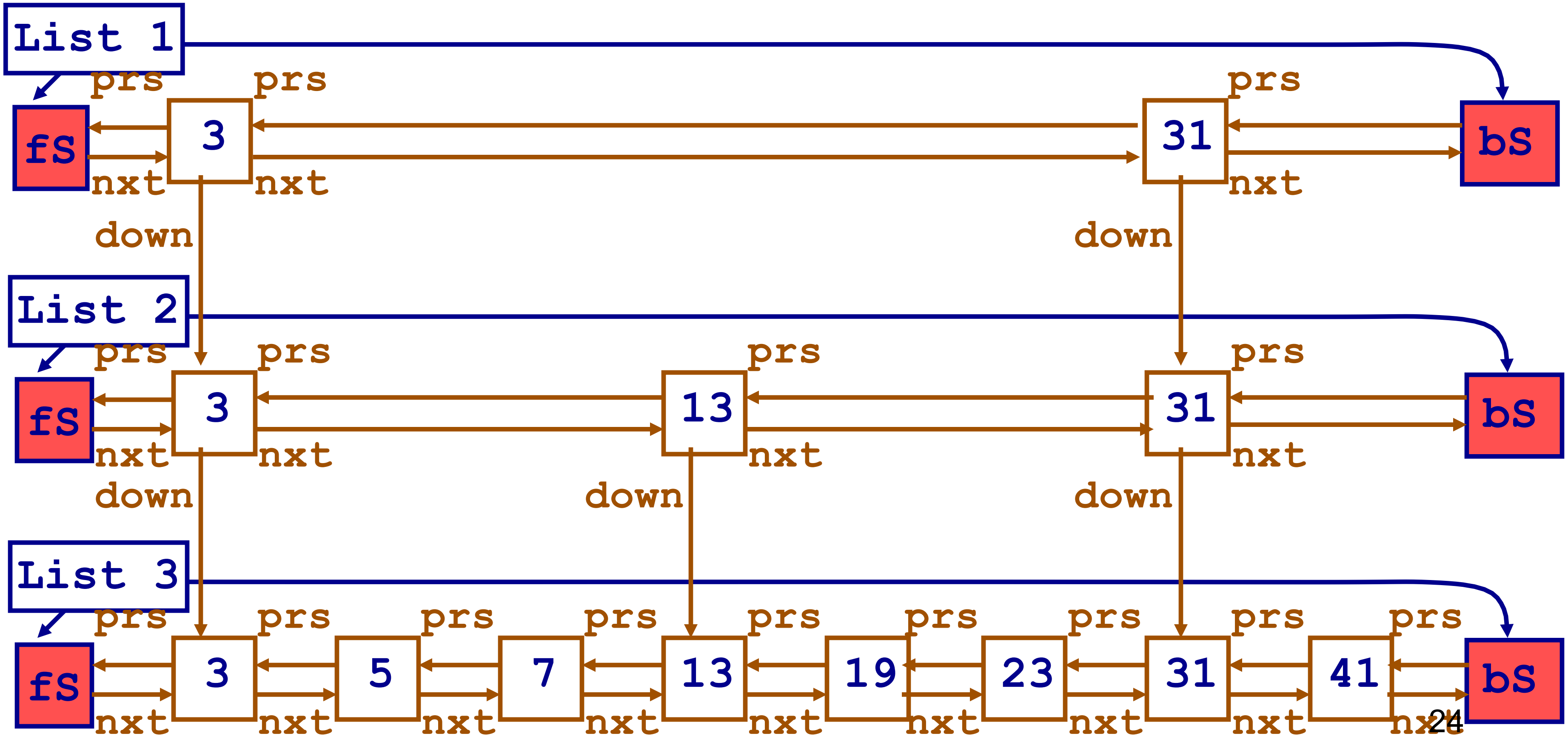
What is Complexity for 3 Lists?

$$|L_1| + \frac{|L_2|}{|L_1|} + \frac{|L_3|}{|L_2|} = 3\sqrt[3]{n}$$



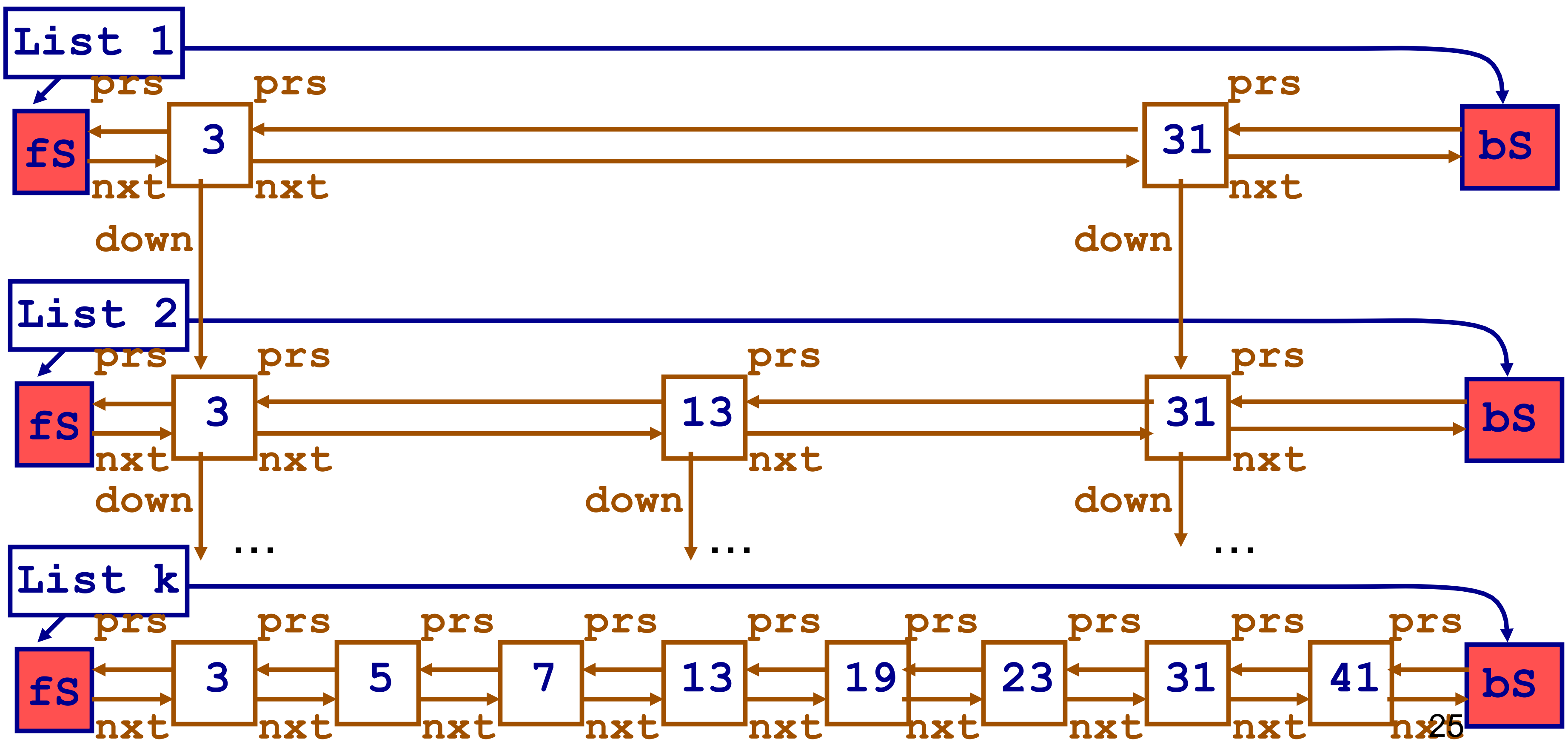
What is Complexity for 3 Lists?

3 lists		2 lists		single list
$O(\sqrt[3]{n})$	$<$	$O(\sqrt{n})$	$<$	$O(n)$



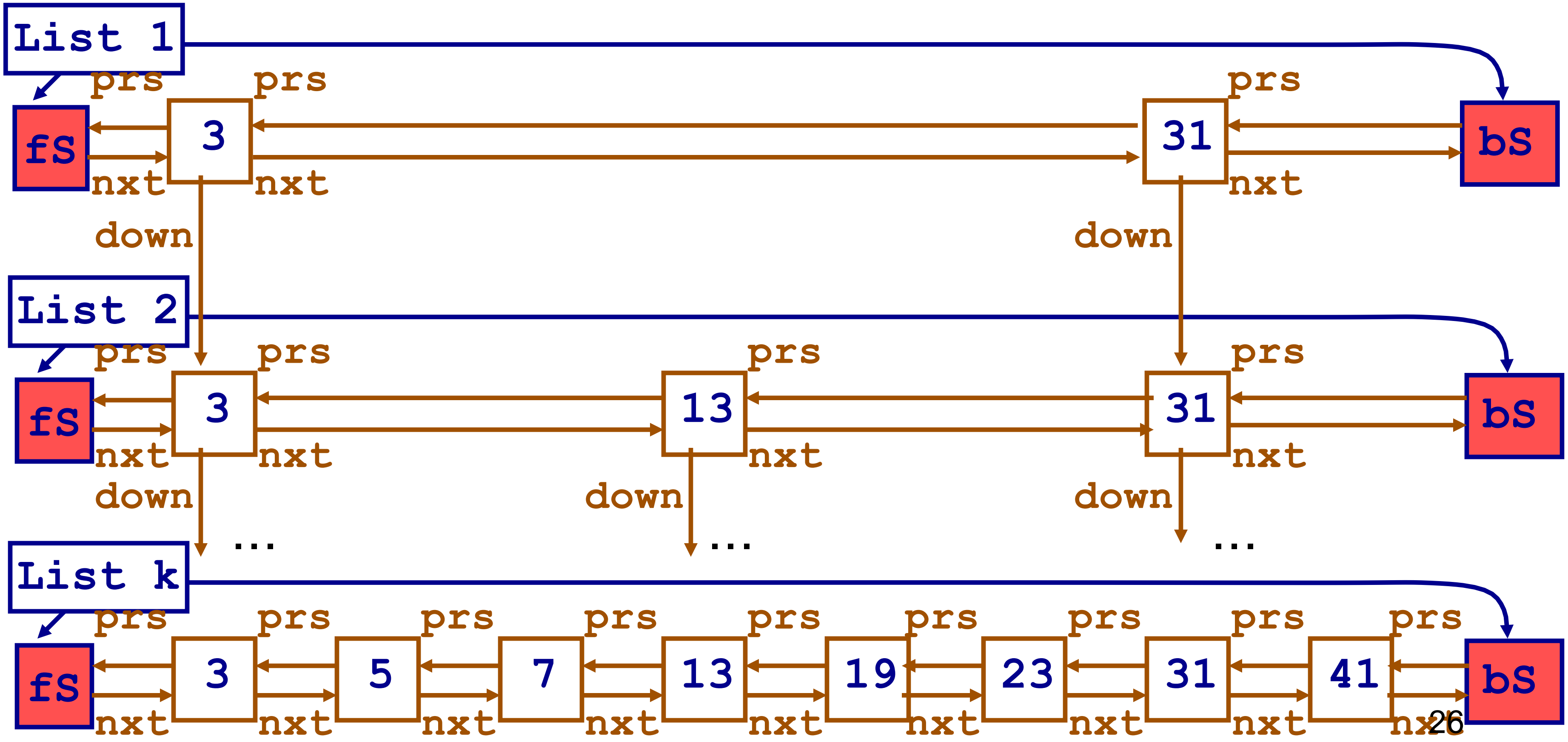
What is Complexity for k Lists?

for k linked lists: $k\sqrt[k]{n}$



How Many Lists?

minimize $k\sqrt[k]{n}$



How Many Lists?

$$\underset{k}{\text{minimize}} \; k \, n^{\frac{1}{k}}$$

=

$$\underset{k}{\text{minimize}} \; \log\left(k \, n^{\frac{1}{k}}\right)$$

How Many Lists?

$$\underset{k}{\text{minimize}} \quad k \; n^{\frac{1}{k}}$$

=

$$\underset{k}{\text{minimize}} \quad \left(\log k + \frac{1}{k} \log n \right)$$

How Many Lists?

$$\frac{d\left(\log k + \frac{1}{k}\log n\right)}{dk} = 0$$

$$\parallel$$

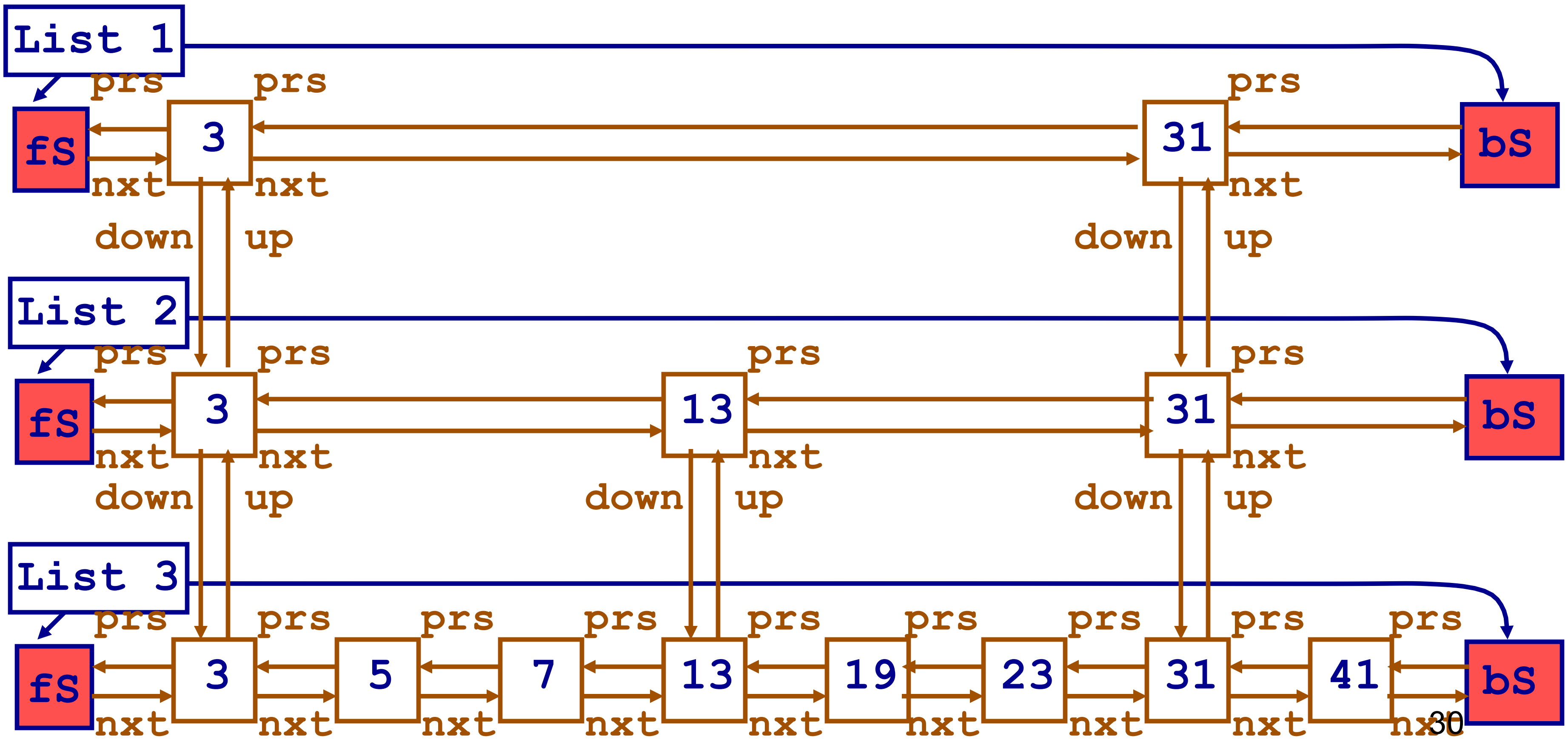
$$\frac{1}{k} - \frac{1}{k^2}\log n = 0$$

$$\Downarrow$$

$$k = \log n$$

Complexity for k=log n Lists?

$$k \; n^{\frac{1}{k}} = (\log n) \; n^{\frac{1}{\log n}}$$



$$n = 2^{\log n}$$

$$\Downarrow$$

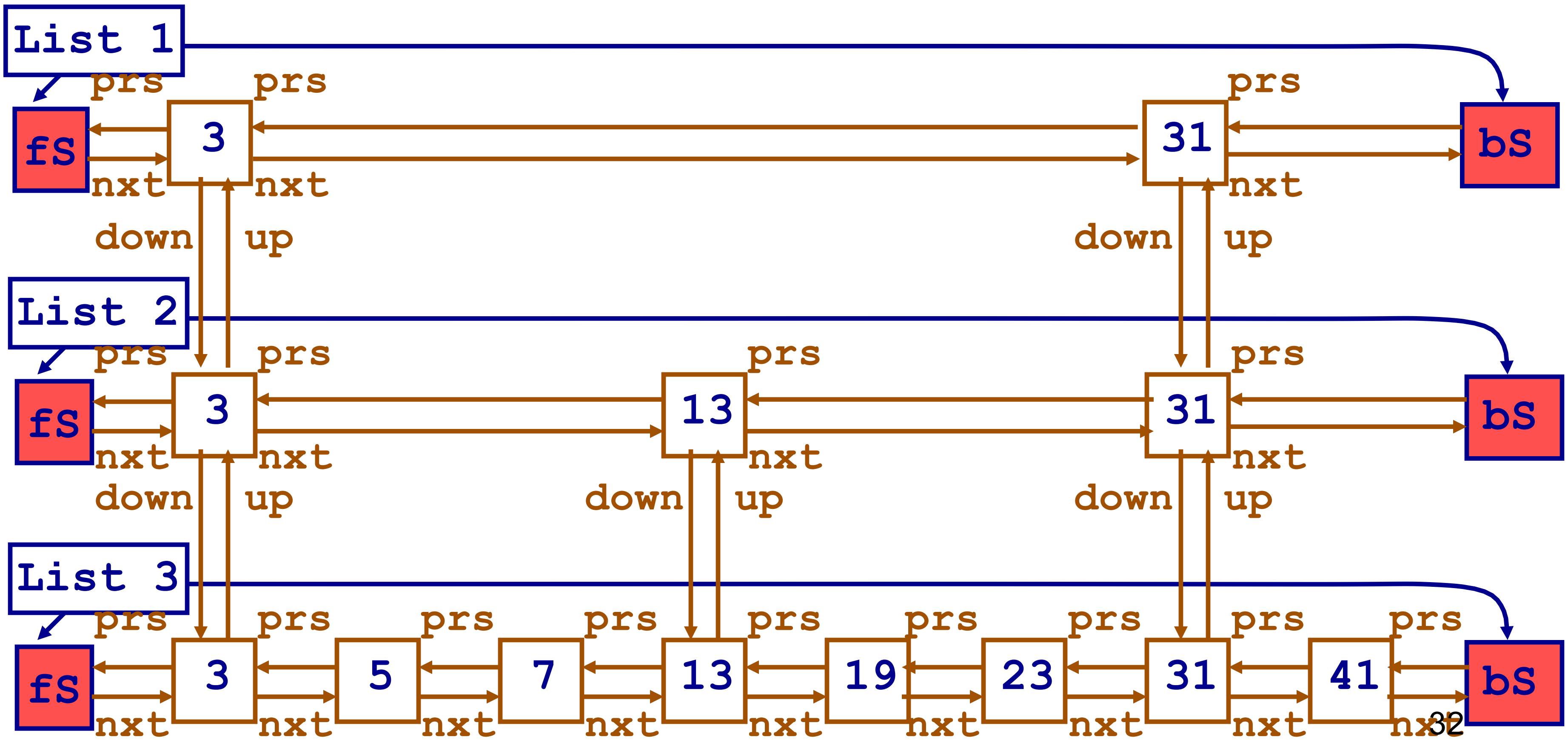
$$n^{\frac{1}{\log n}} = \left(2^{\log n}\right)^{\frac{1}{\log n}} = 2$$

$$\Downarrow$$

$$(\log n)\ n^{\frac{1}{\log n}} = 2 \log n$$

Complexity for log n Lists

$$O(\log n)$$



Skip Lists – Have It All
Pugh 1989

- Fast addition O(log n)
- Fast search O(log n)
- Fast removal O(log n)

- Disadvantage: - *Slightly* more complicated

Contains Skip List

- Makes a zig-zag motion top-to-bottom
- Complexity: O(log n), i.e., proportional to the number of linked lists

Contains Skip List

1. Start at topmost sentinel
2. Loop as follows
   1. Slide right, get a link right before
   2. If next element is OK, return true
   3. If no down element, return false
   4. Move down

Remove Skip List

- Makes a zig-zag motion top-to-bottom
- Only decrement the size at the bottom level
- Complexity: O(log n), i.e., proportional to the number of linked lists

Remove Skip List

1. Start at topmost sentinel
2. Loop as follows
   1. Slide right, get a link right before
   2. If next element is OK, remove it
   3. If no down element, reduce size
   4. Move down

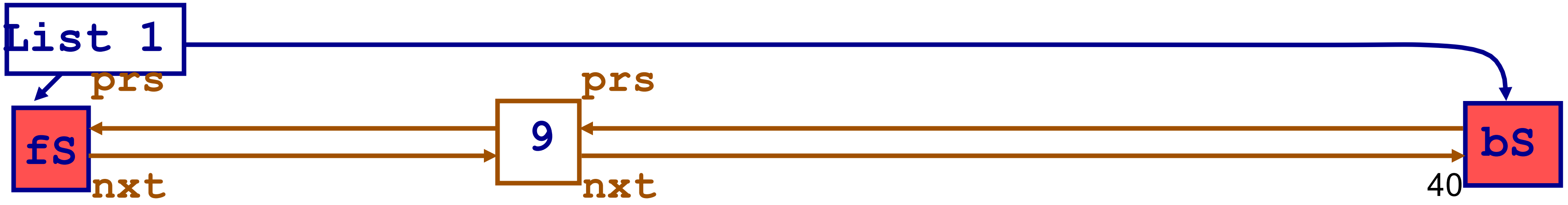
How to construct a skip list when we do not know the number of elements in advance?

Add Skip List

- Add the element to the bottom list
  - must increment size
- To move up to existing lists:
  - Flip a coin, and if heads add the element up
- To add a new list at the top
  - Flip a coin, and if heads make a new top list

Add Example

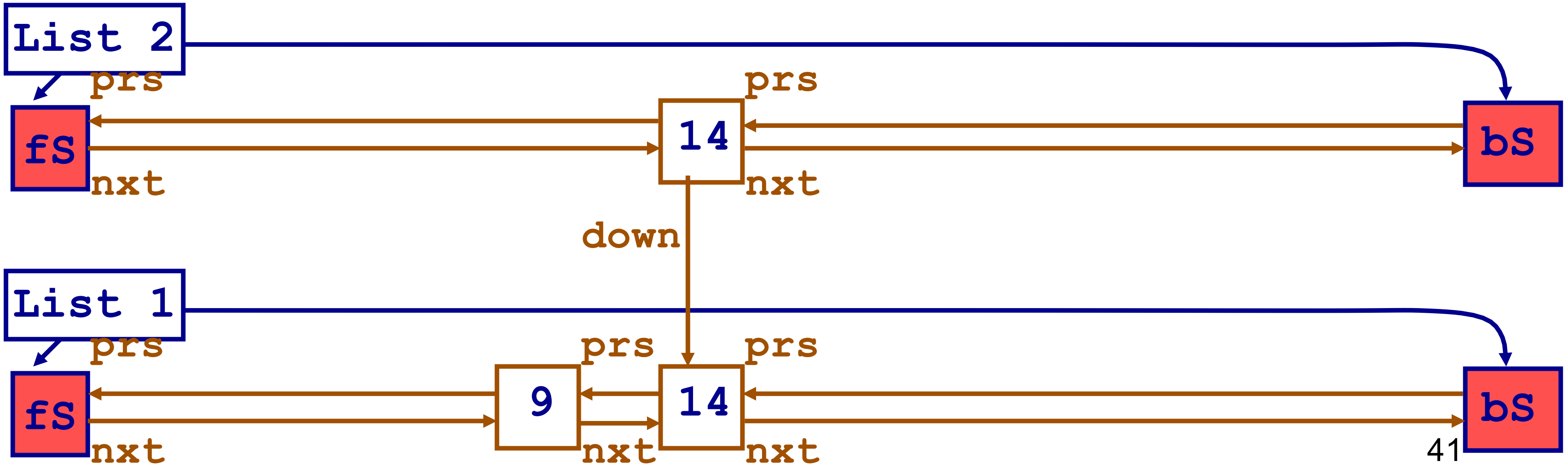
Insert the following: 9 14 7 3 20



Add Example:

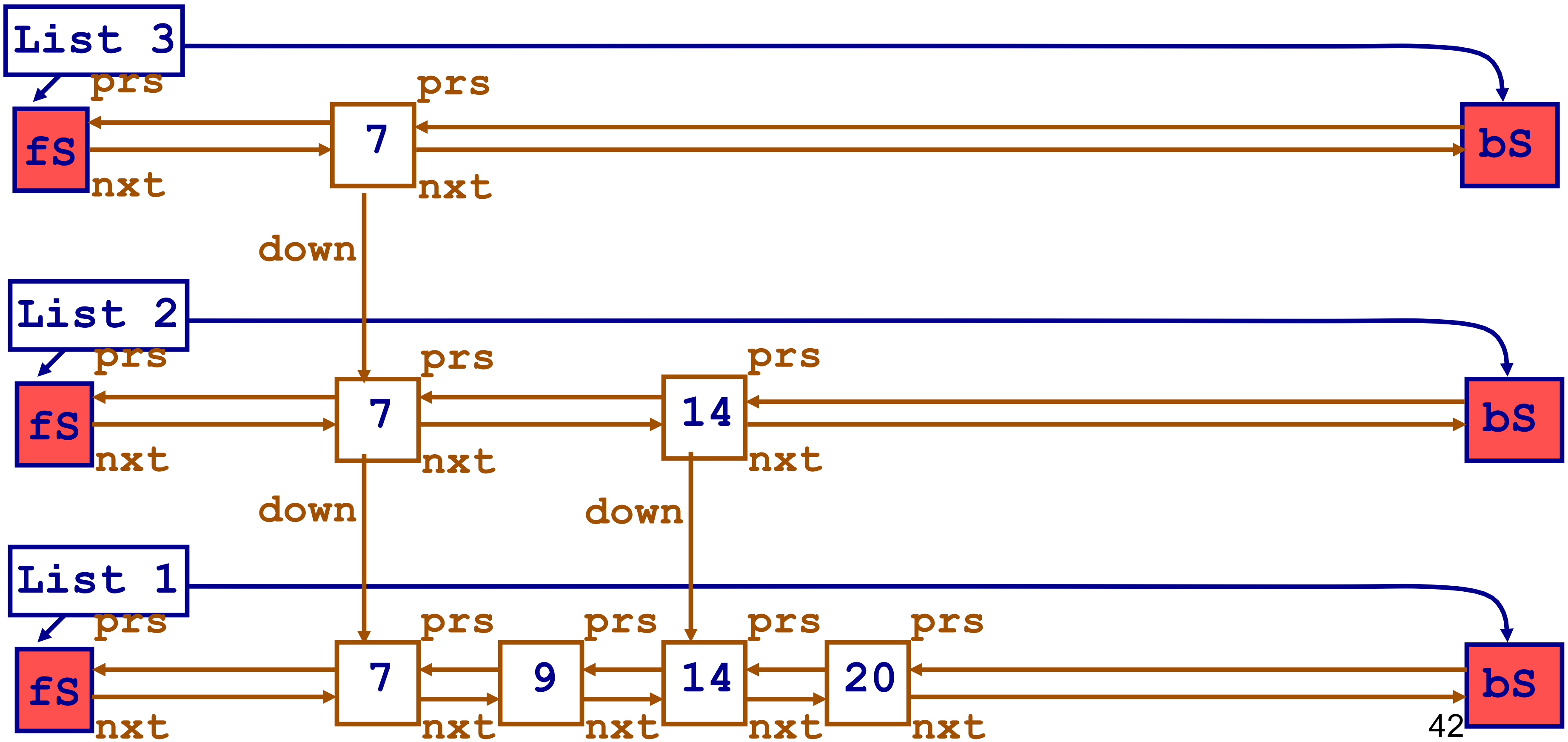
Inserted : 9 14

Coin toss: T HT (H = move up)



Add Example:

Inserted :	9	14	7	20
Coin toss:	T	HT	HHH	T



Complexity of Add

- Proportional to the height, not to the number of nodes in the list
- O(log n)

Skip List Sorting Algorithm

Problem: Sort an array A

Step 1. Copy elements from A into a skip list

Step 2. Copy elements from the skip list to A

Skip List Sorting Algorithm

Complexity:

Step 1. Copy elements from A into a skip list

O(??)

Step 2. Copy elements from the skip list to A

O(??)