

Skip List

Shusen Wang

<http://wangshusen.github.io/>

Why skip list?

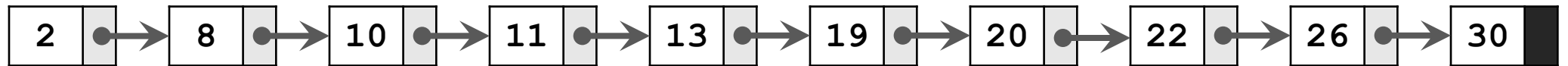
- Linked list does not support binary search.
- Skip list allows fast search and fast insertion.
- **Search:** $O(\log n)$ time complexity on average.
- **Insertion:** $O(\log n)$ time complexity on average.

Outline

1. Building a skip list.
2. Search.
3. Insertion.

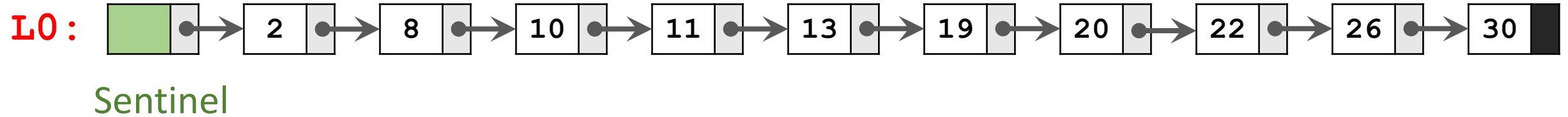
Building a Skip List

Initial State

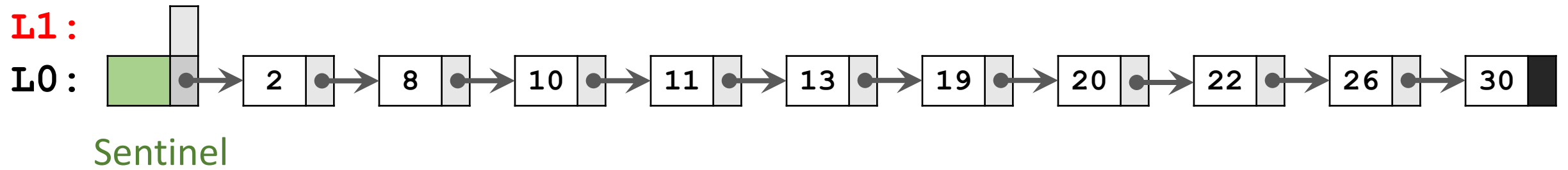


Initially, a linked list contains n numbers in ascending order.

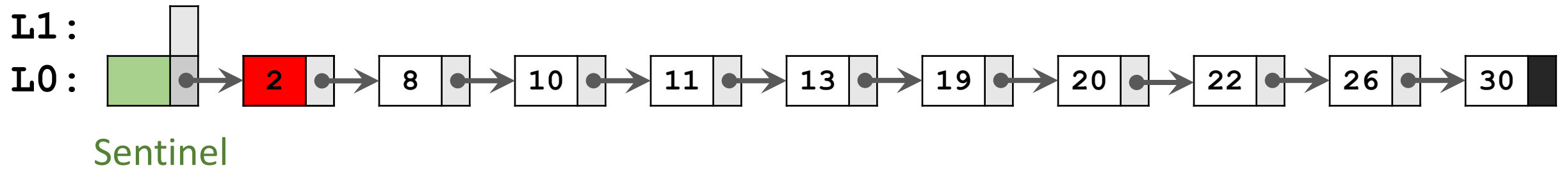
Add sentinel in the front



Iteration 1

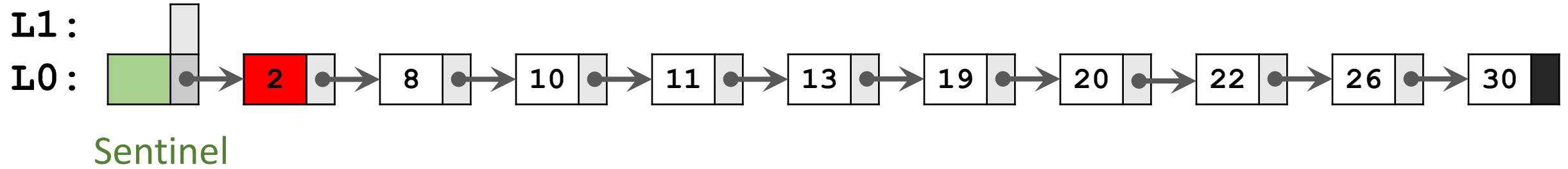


Iteration 1(A)



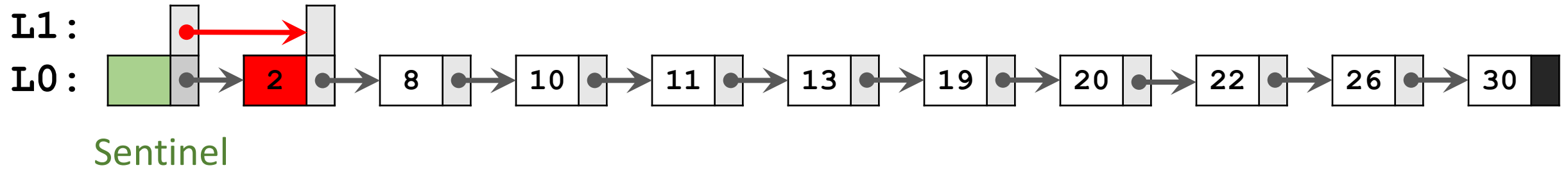
Iteration 1(A)

Flip a coin.



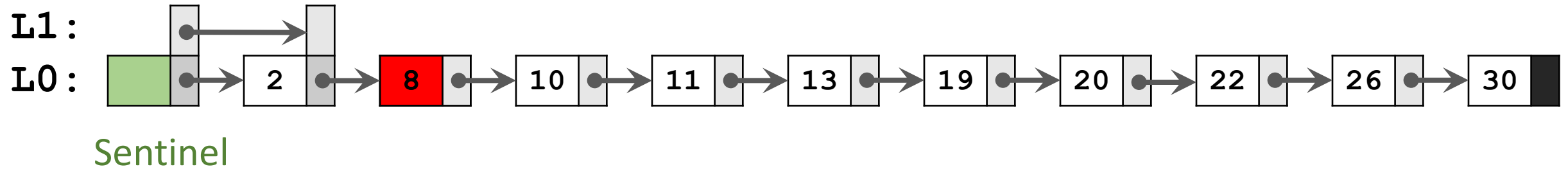
Iteration 1(A)

Flip a coin.



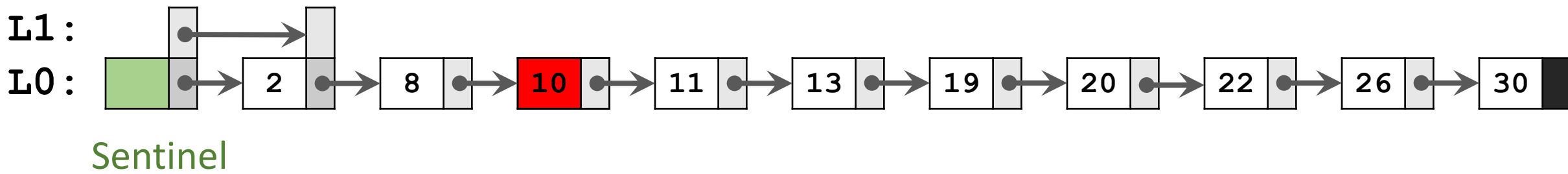
Iteration 1(B)

Flip a coin.



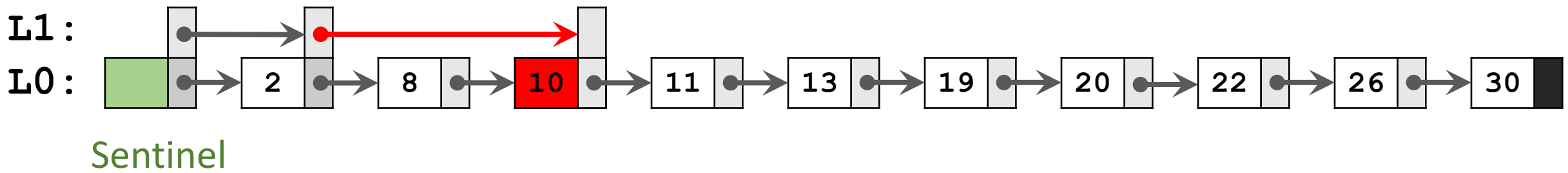
Iteration 1(C)

Flip a coin.



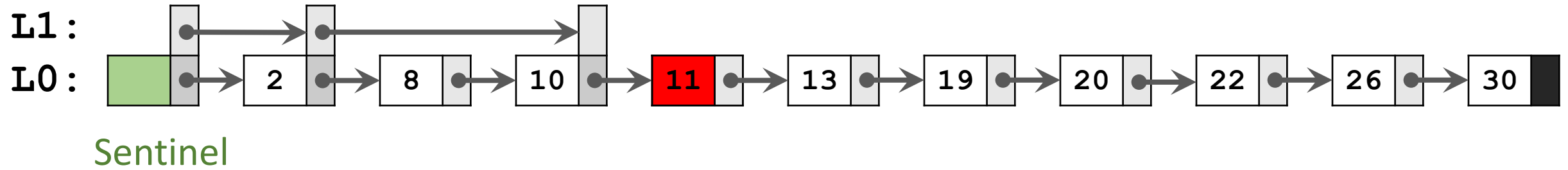
Iteration 1(C)

Flip a coin.



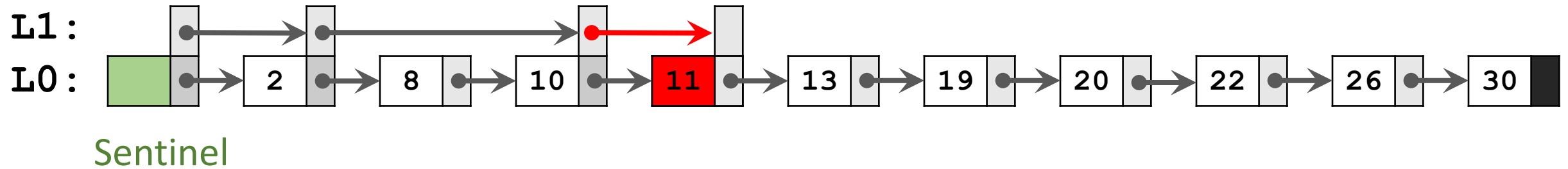
Iteration 1(D)

Flip a coin.



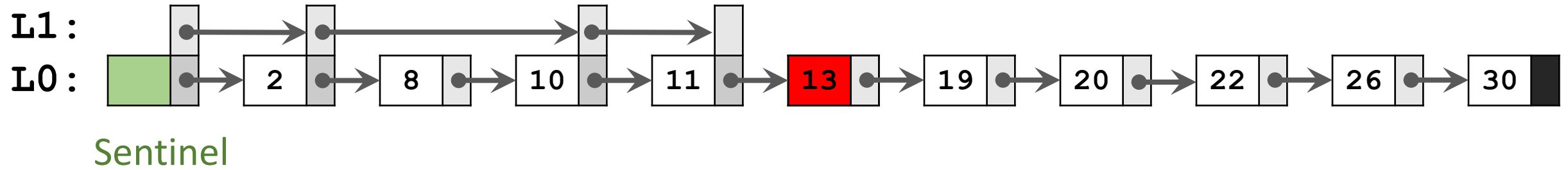
Iteration 1(D)

Flip a coin.



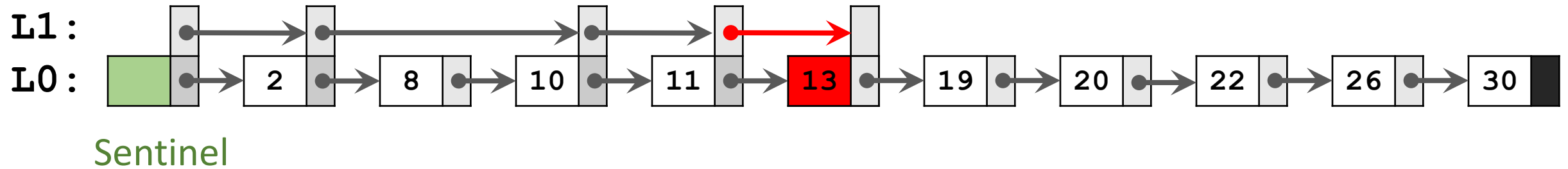
Iteration 1(E)

Flip a coin.



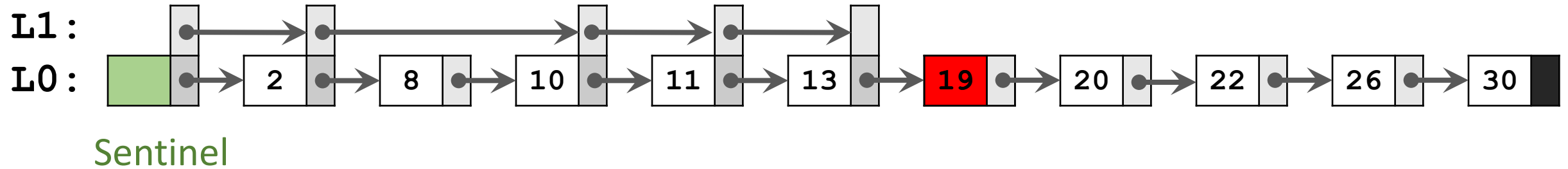
Iteration 1(E)

Flip a coin.



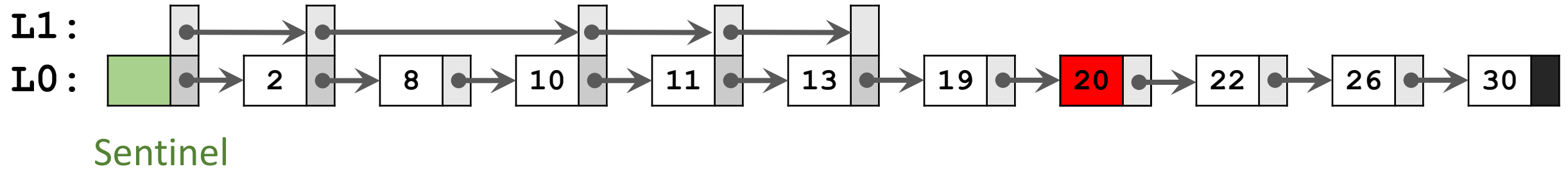
Iteration 1(F)

Flip a coin.



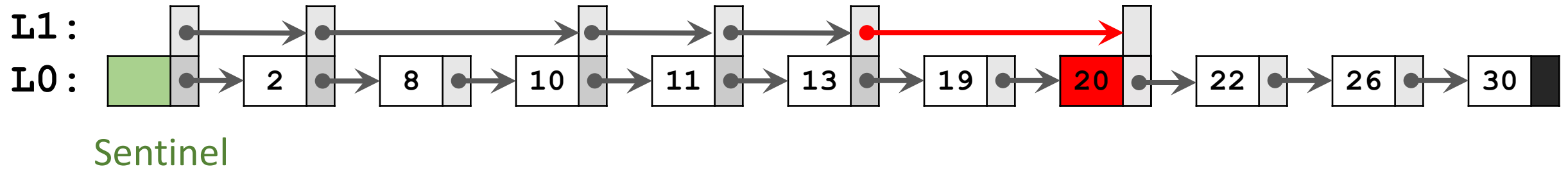
Iteration 1(G)

Flip a coin.



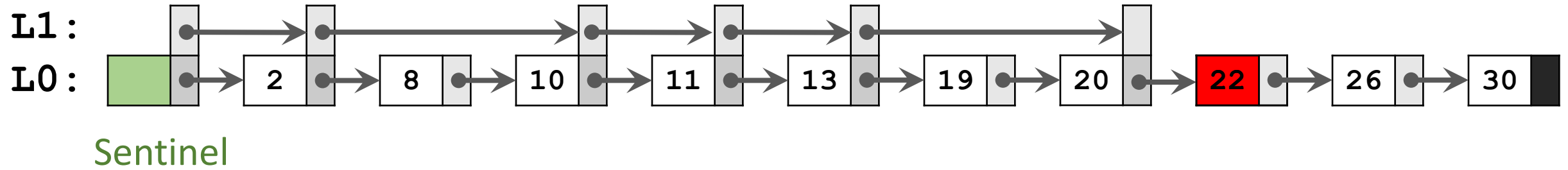
Iteration 1(G)

Flip a coin.



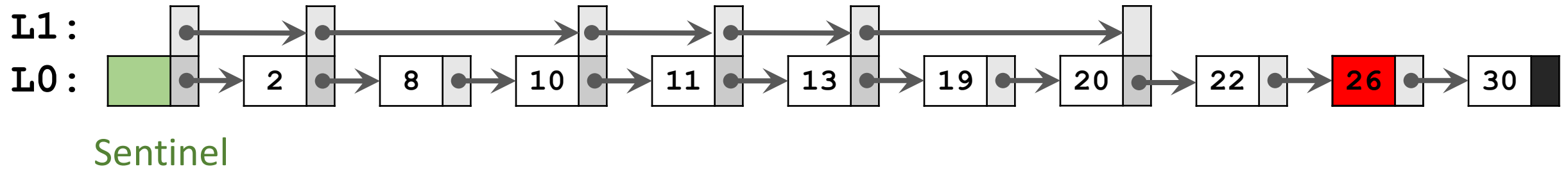
Iteration 1(H)

Flip a coin.



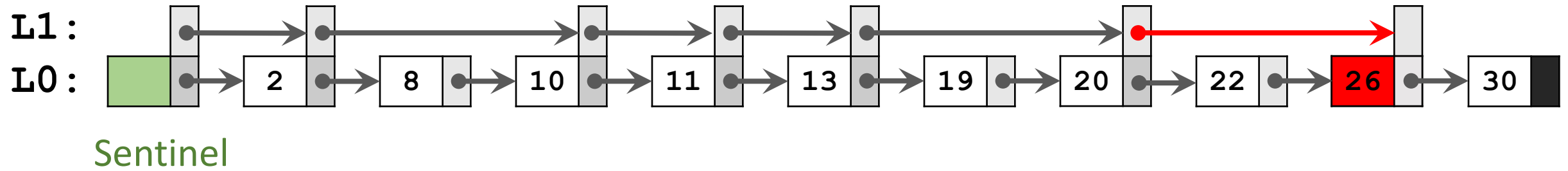
Iteration 1(I)

Flip a coin.



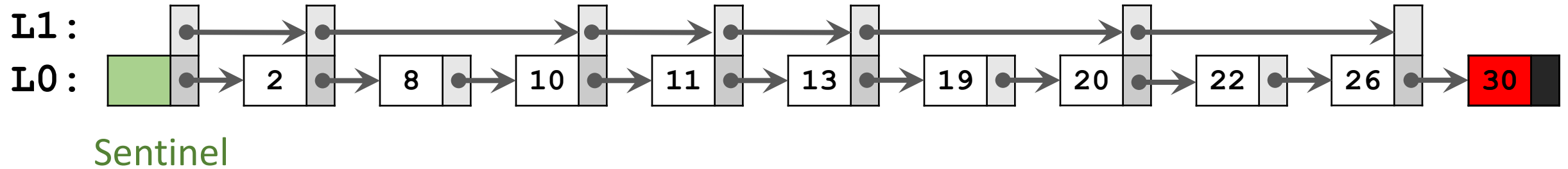
Iteration 1(I)

Flip a coin.

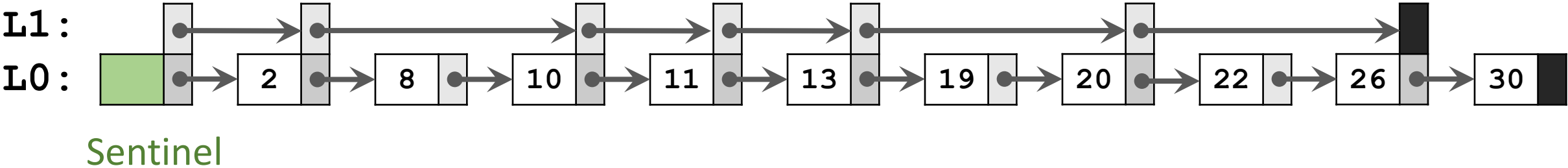


Iteration 1(J)

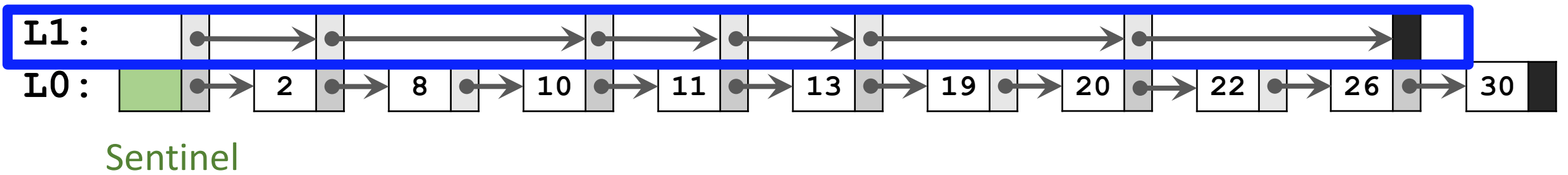
Flip a coin.



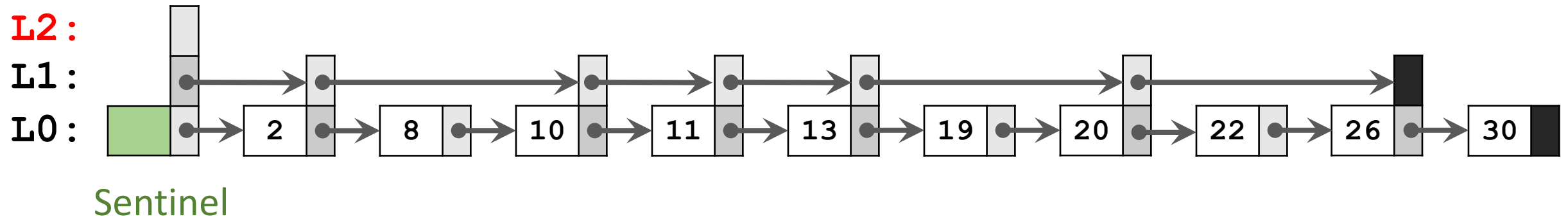
Iteration 1(End)



Iteration 2

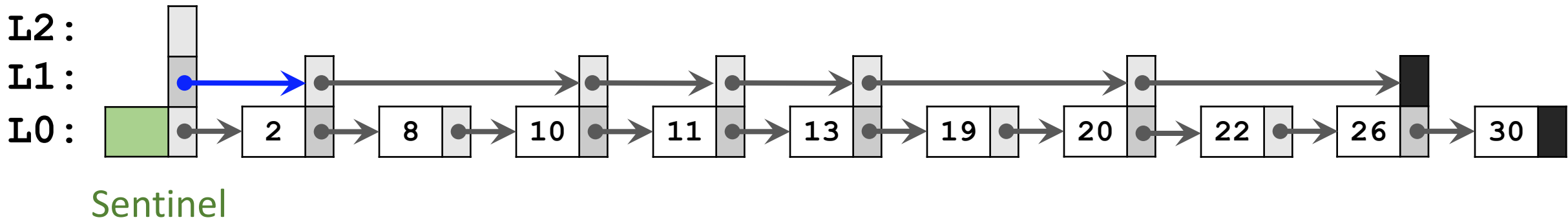


Iteration 2



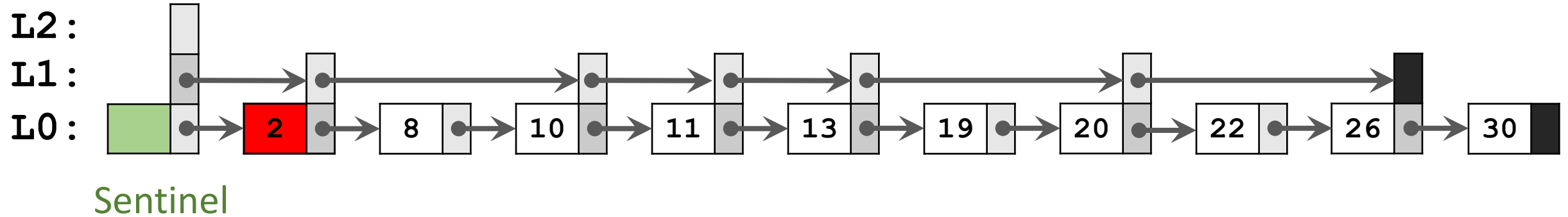
- Build the **L2** linked list.

Iteration 2(A)



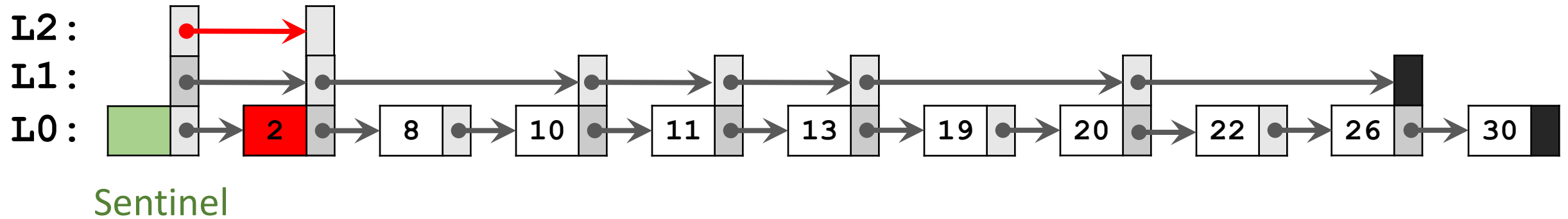
Iteration 2(A)

Flip a coin.

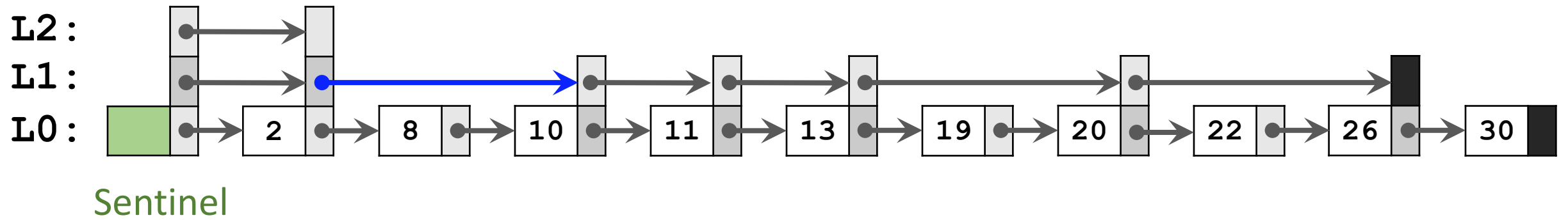


Iteration 2(A)

Flip a coin.

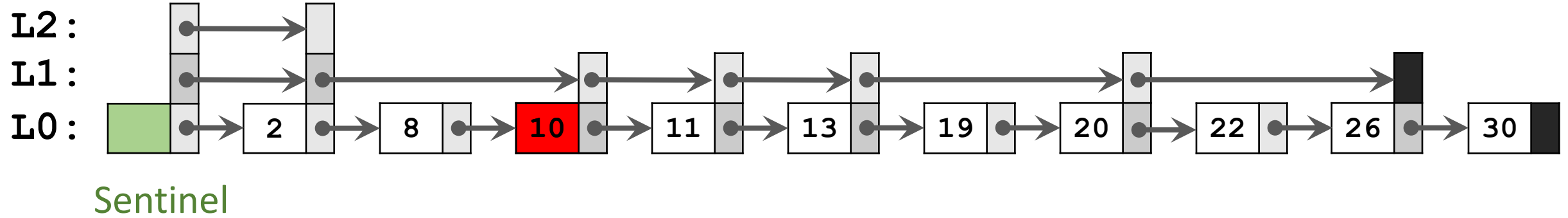


Iteration 2(B)



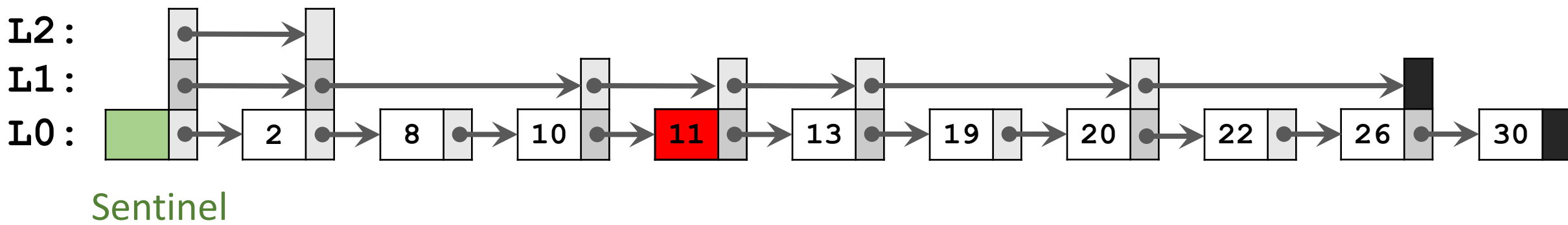
Iteration 2(B)

Flip a coin.



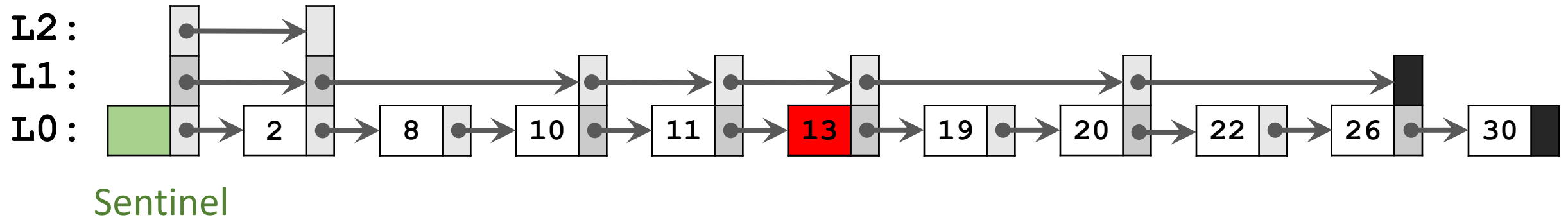
Iteration 2(C)

Flip a coin.



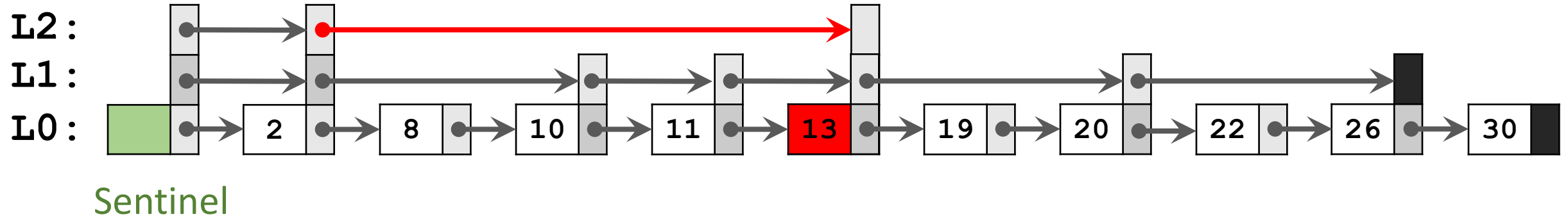
Iteration 2(D)

Flip a coin.



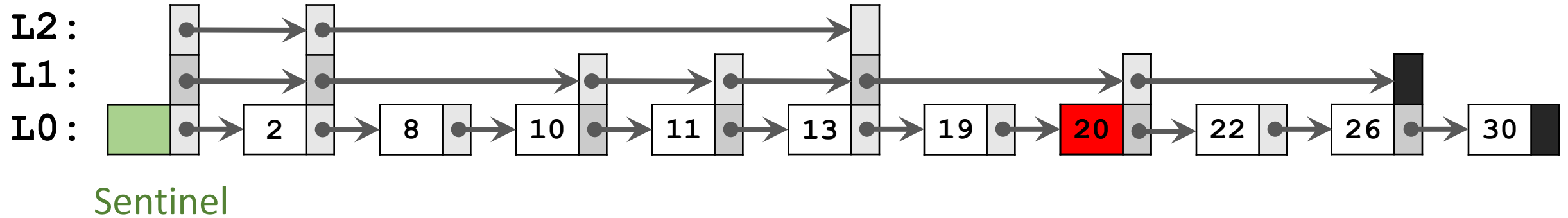
Iteration 2(D)

Flip a coin.



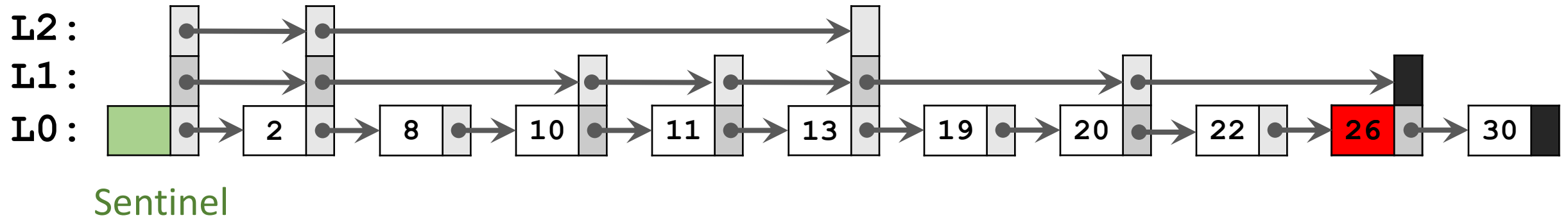
Iteration 2(E)

Flip a coin.



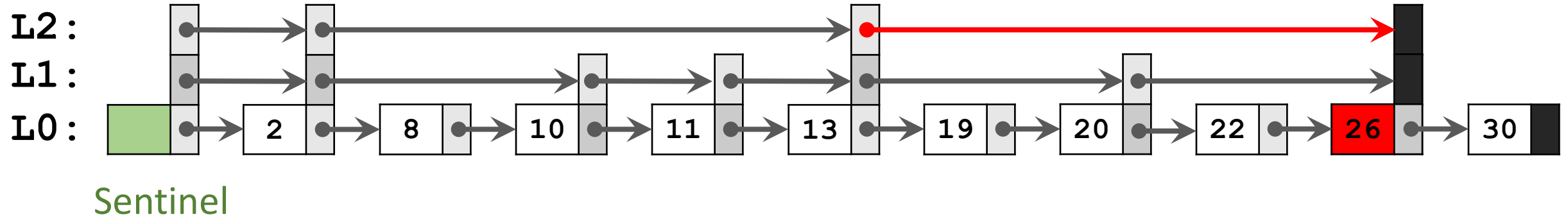
Iteration 2(F)

Flip a coin.

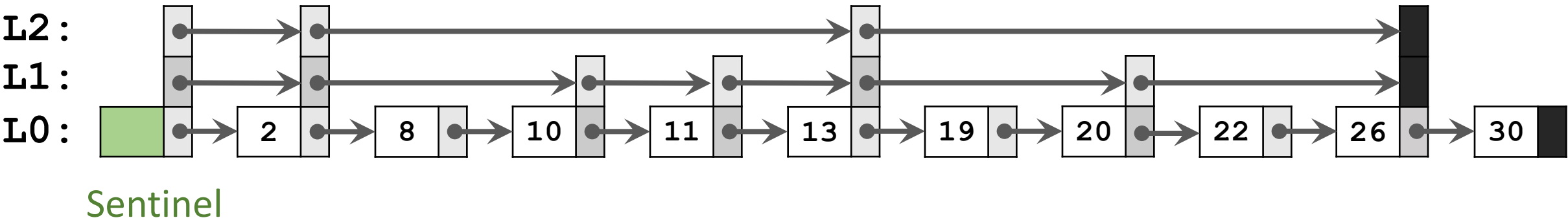


Iteration 2(F)

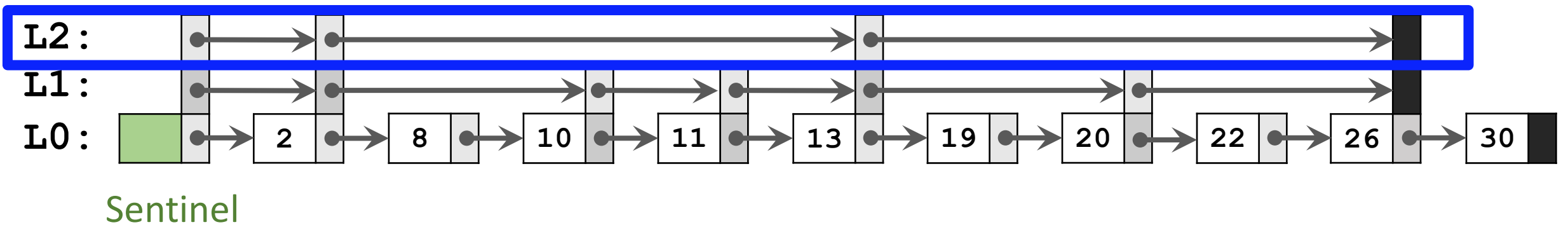
Flip a coin.



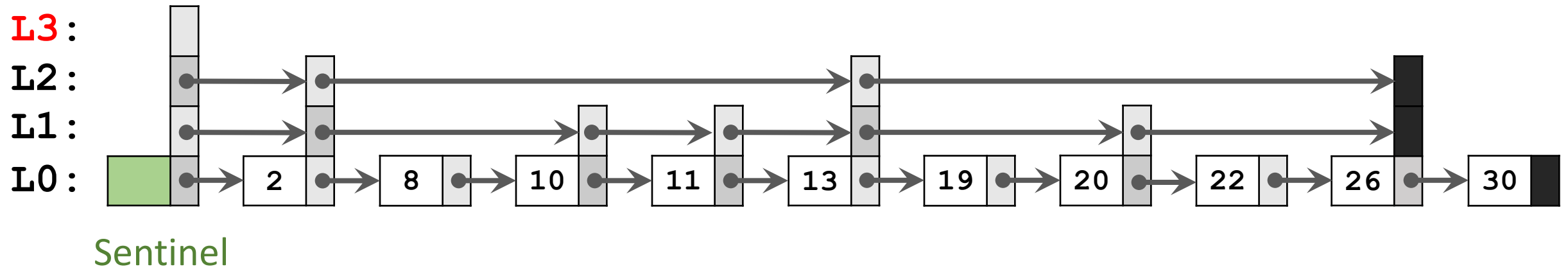
Iteration 2(End)



Iteration 3

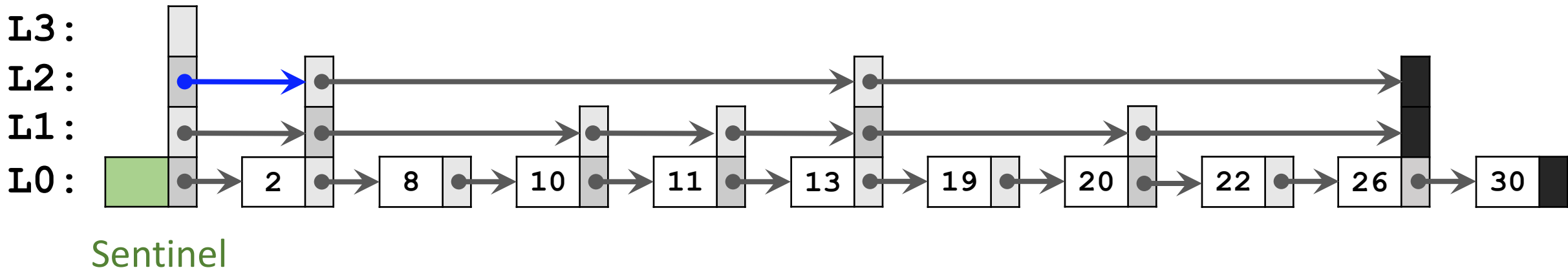


Iteration 3

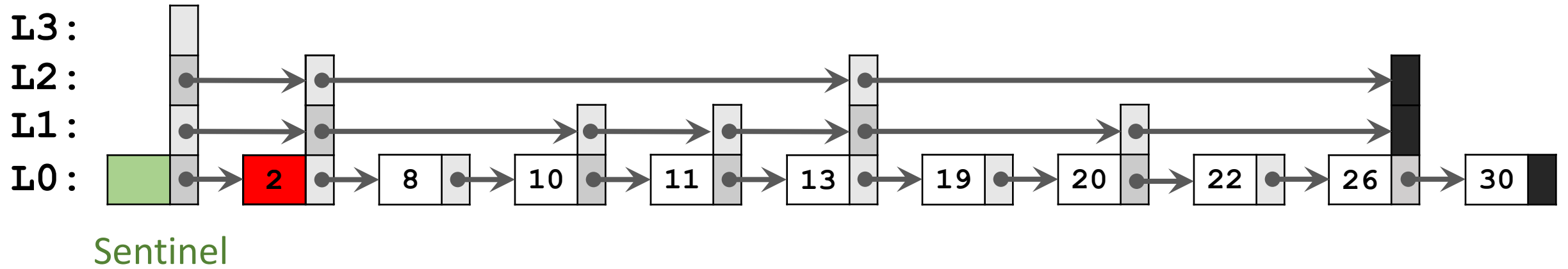


- Build the **L3** linked list.

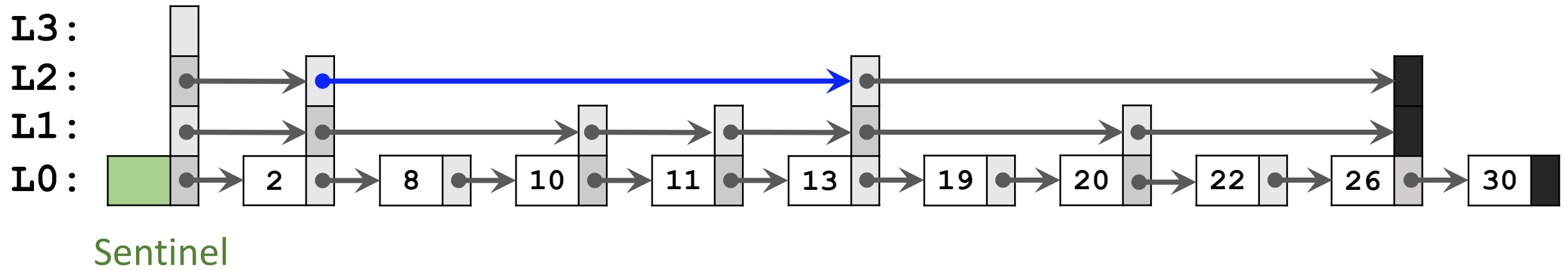
Iteration 3(A)



Flip a coin.

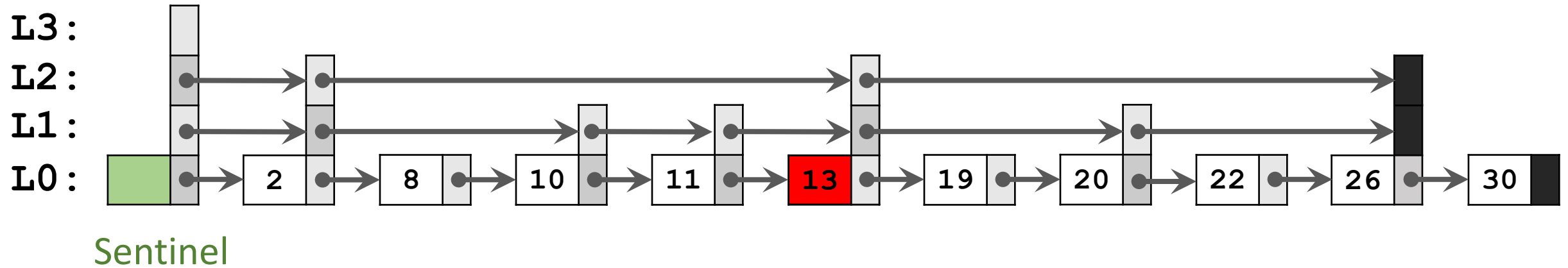


Iteration 3(B)



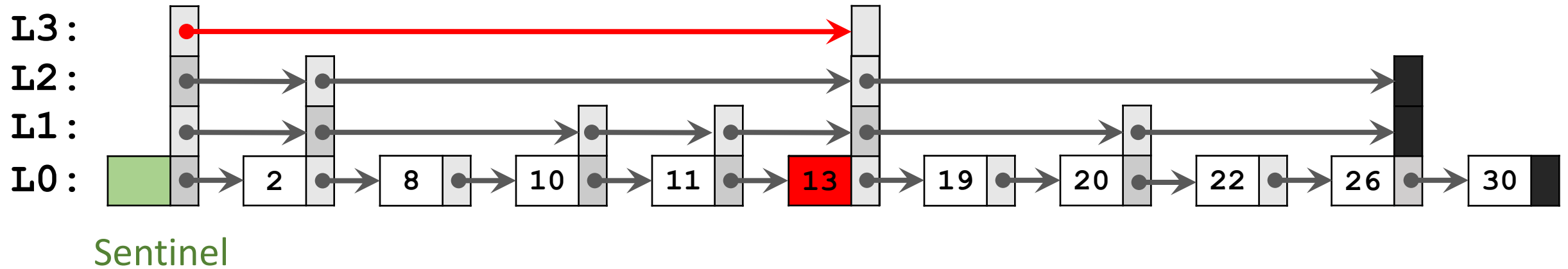
Iteration 3(B)

Flip a coin.

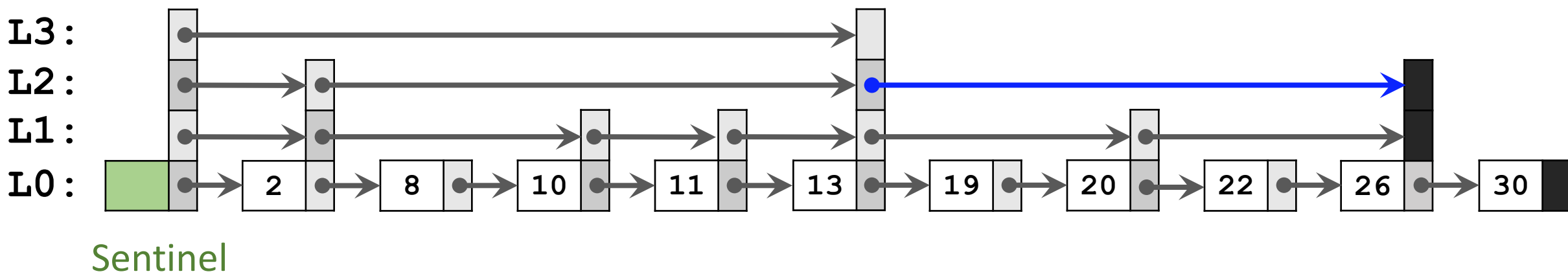


Iteration 3(B)

Flip a coin.

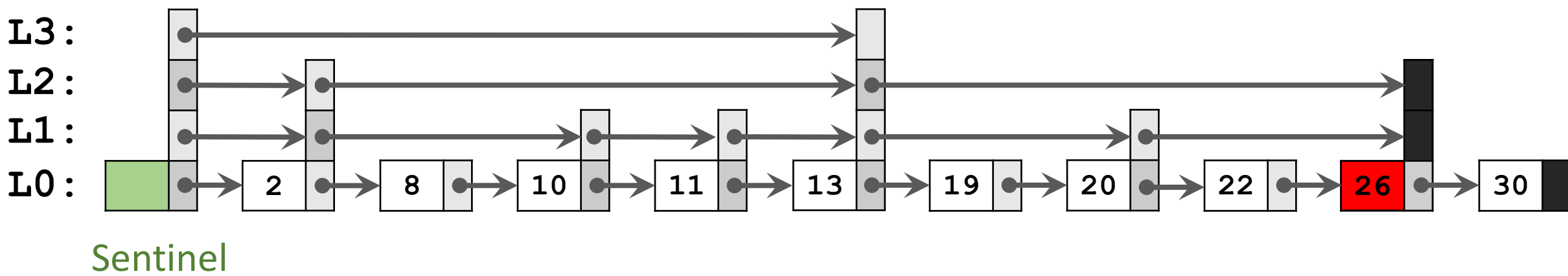


Iteration 3(C)

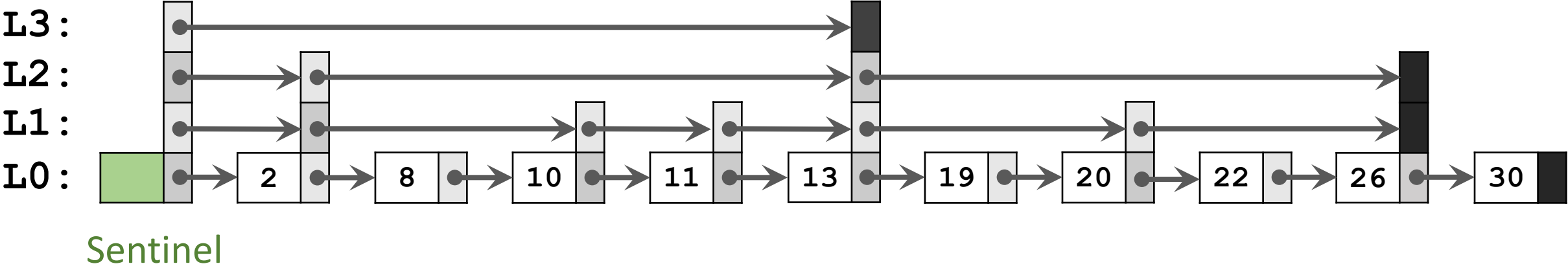


Iteration 3(C)

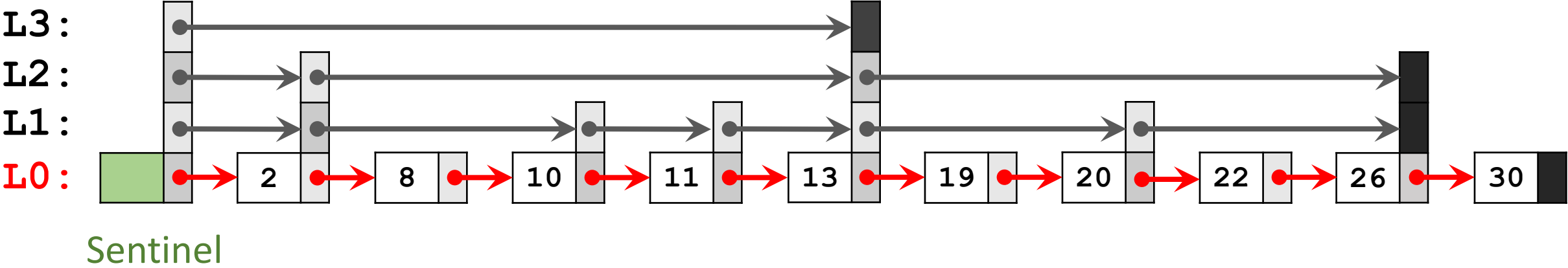
Flip a coin.



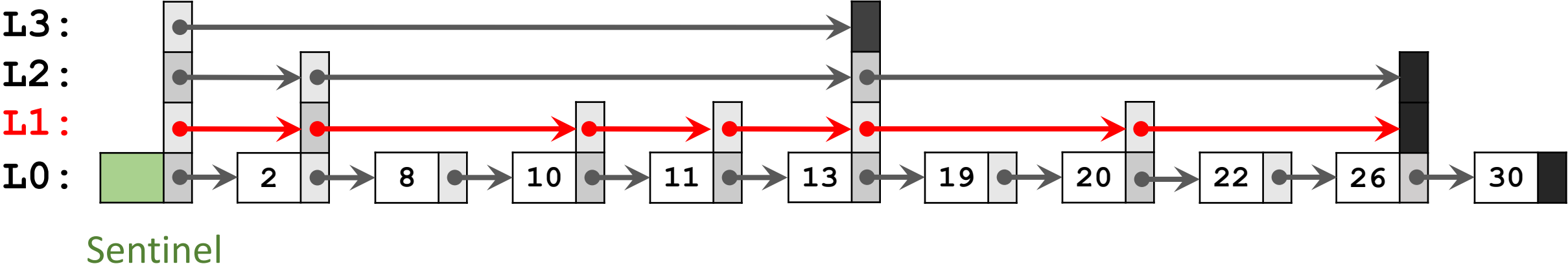
Iteration 3(End)



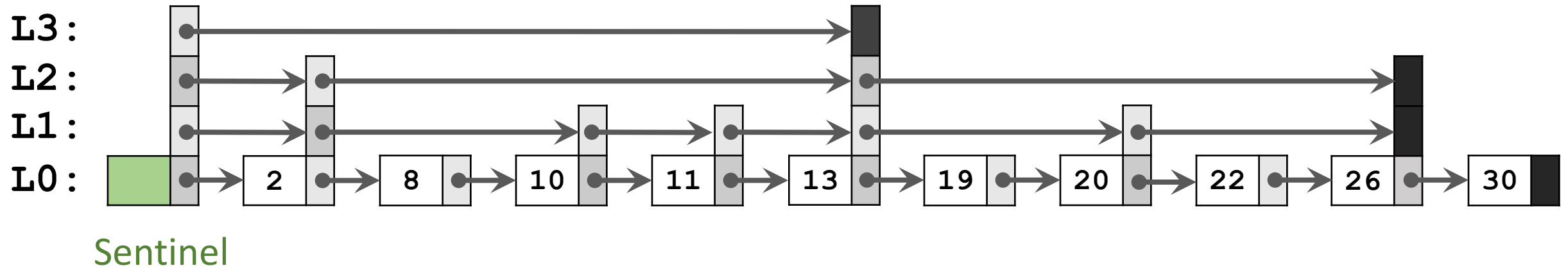
Iteration 3(End)



Iteration 3(End)



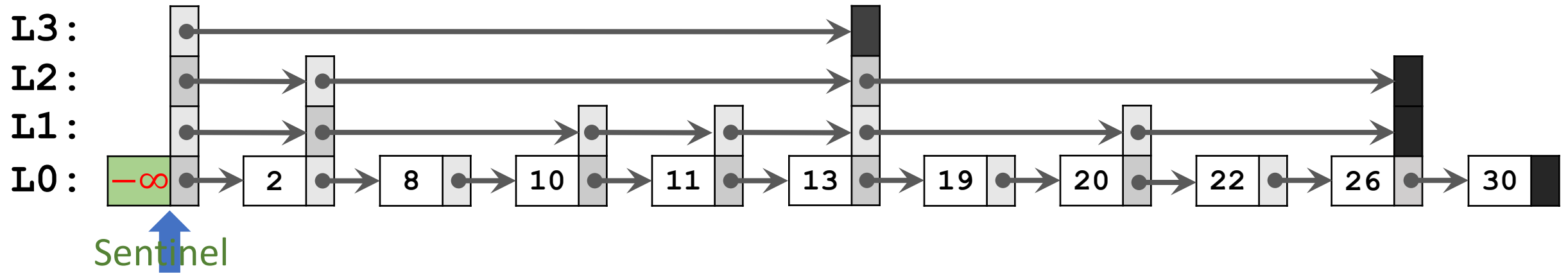
End of Procedure



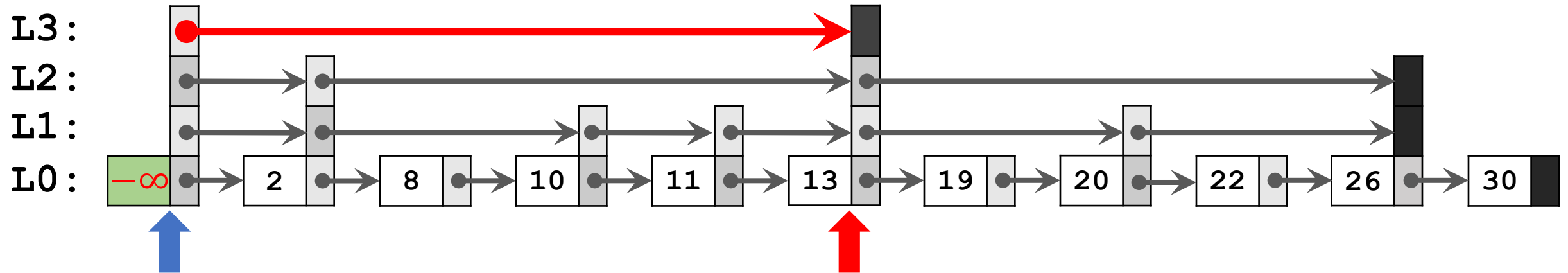
- The number of layers is up to the user; we use a total of 4 layers.
- The number of layers should be $\log n$.

Search

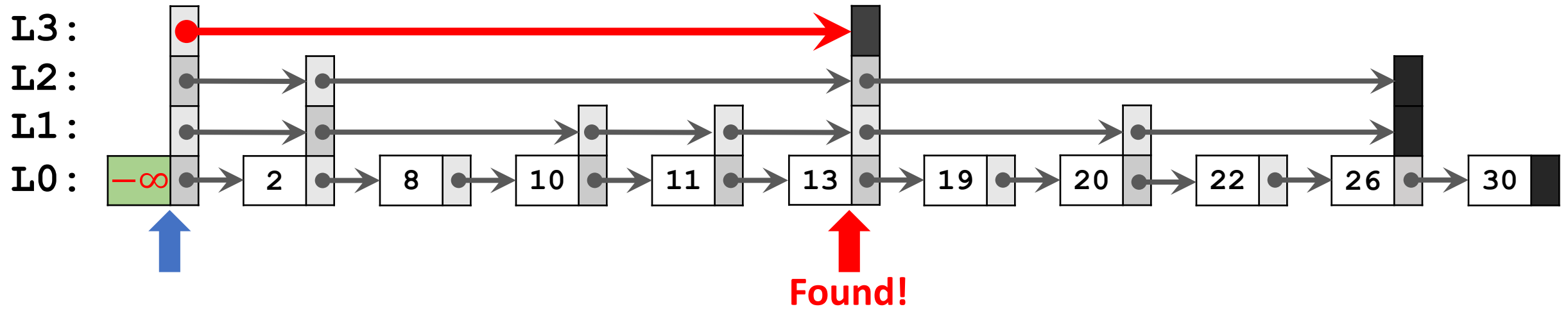
Search: **key=13**



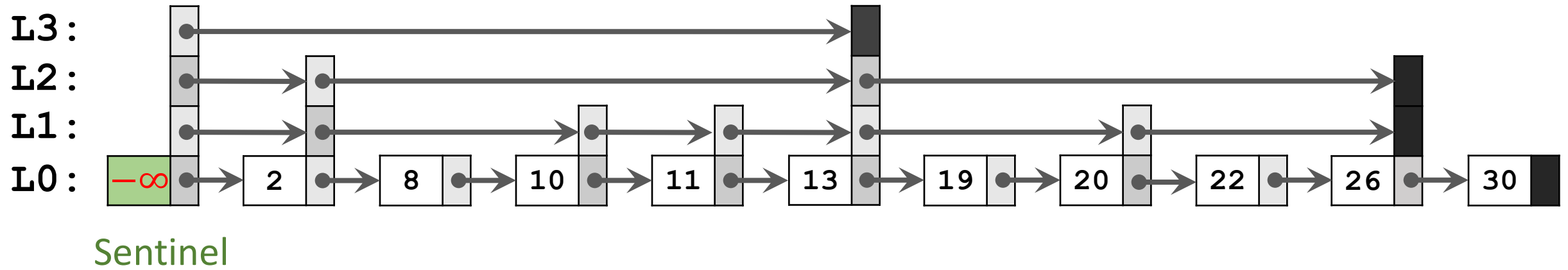
Search: **key=13**



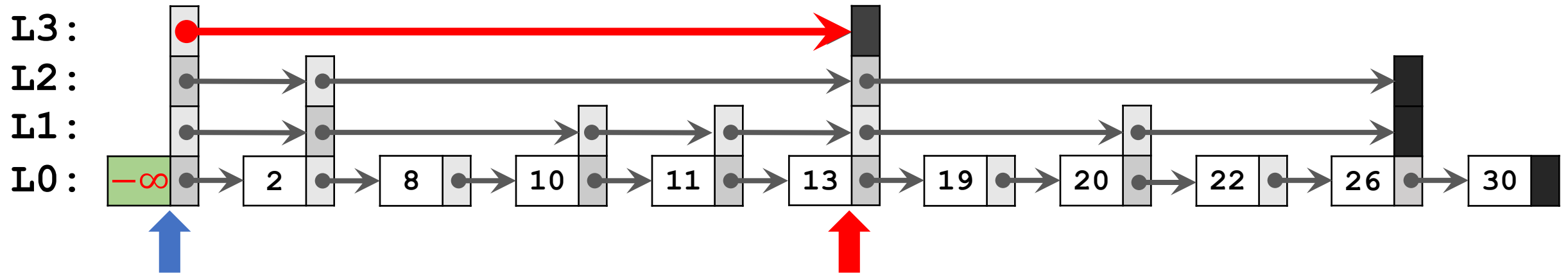
Search: **key=13**



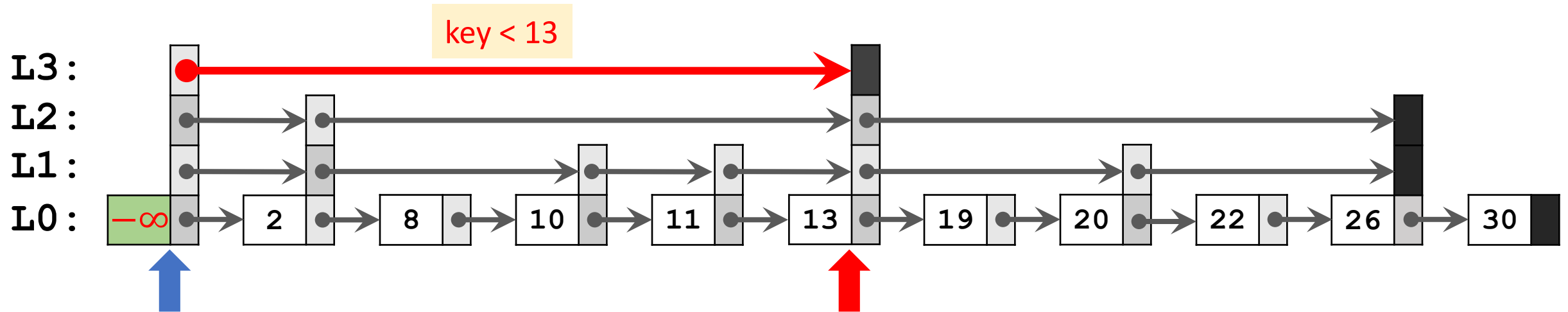
Search: **key=8**



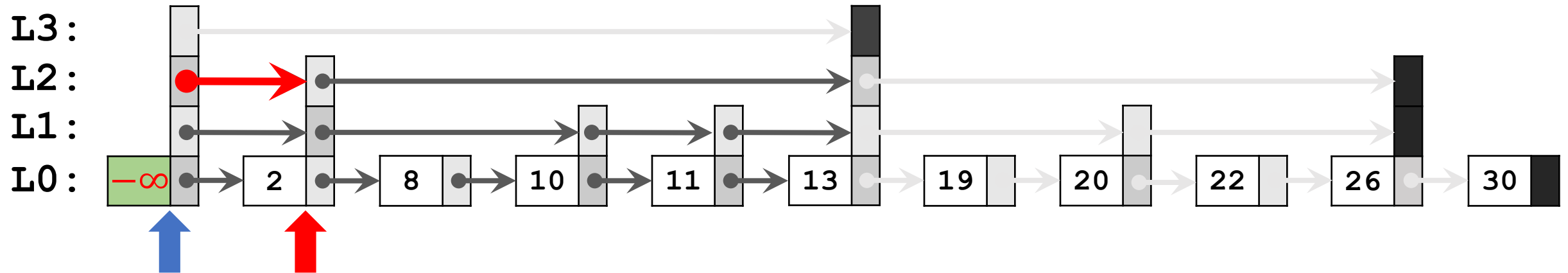
Search: **key=8**



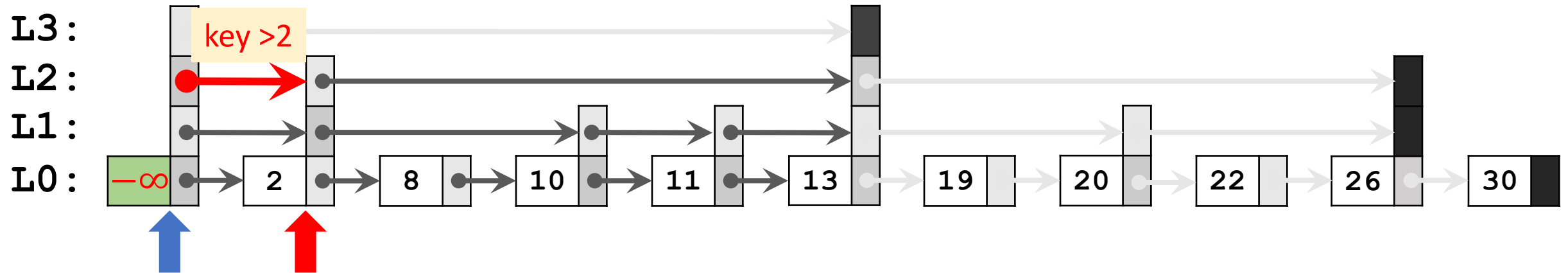
Search: **key=8**



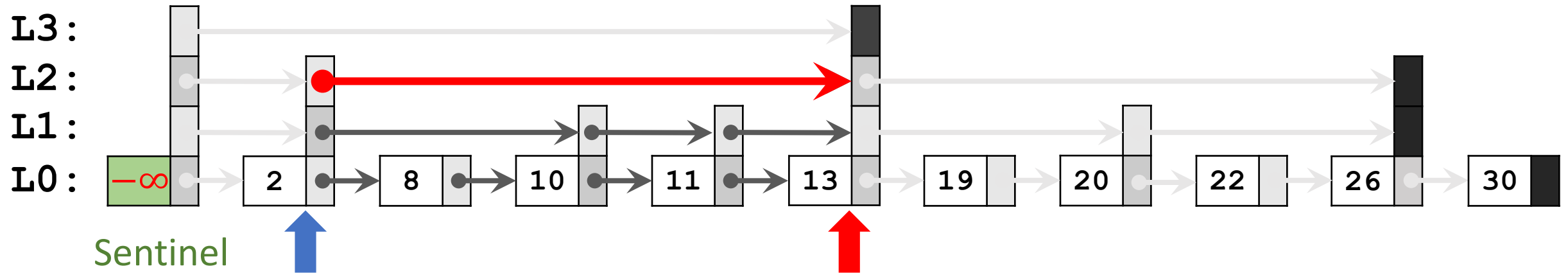
Search: **key=8**



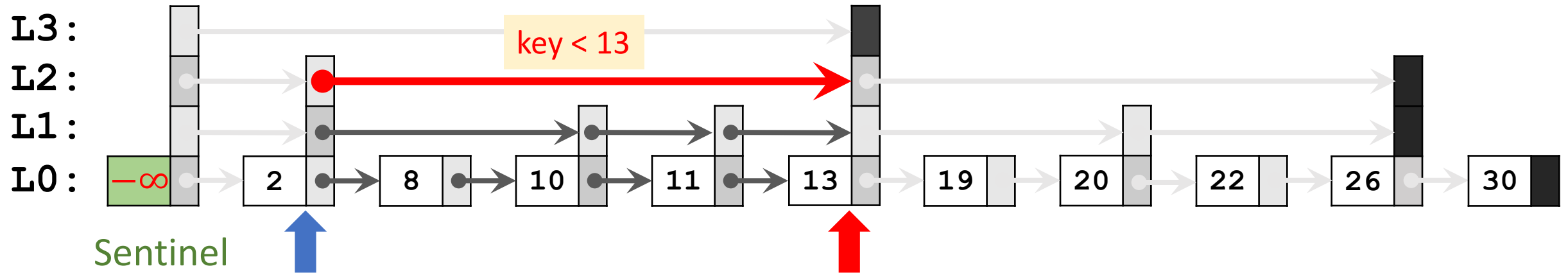
Search: **key=8**



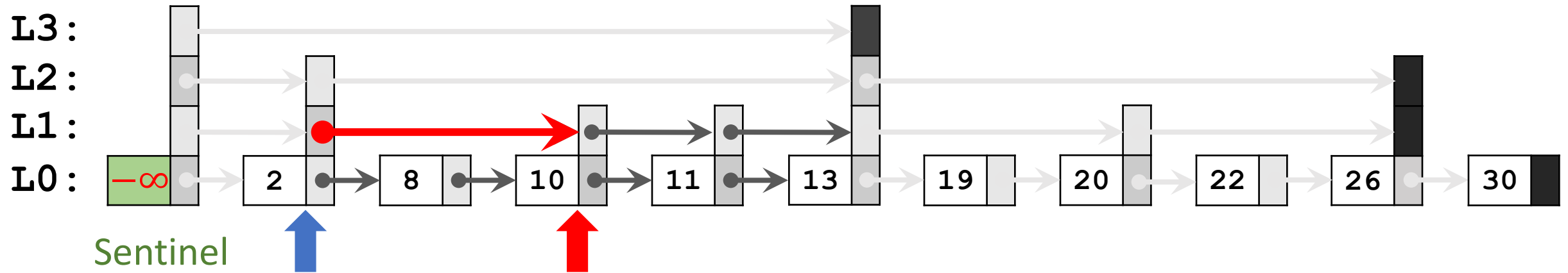
Search: **key=8**



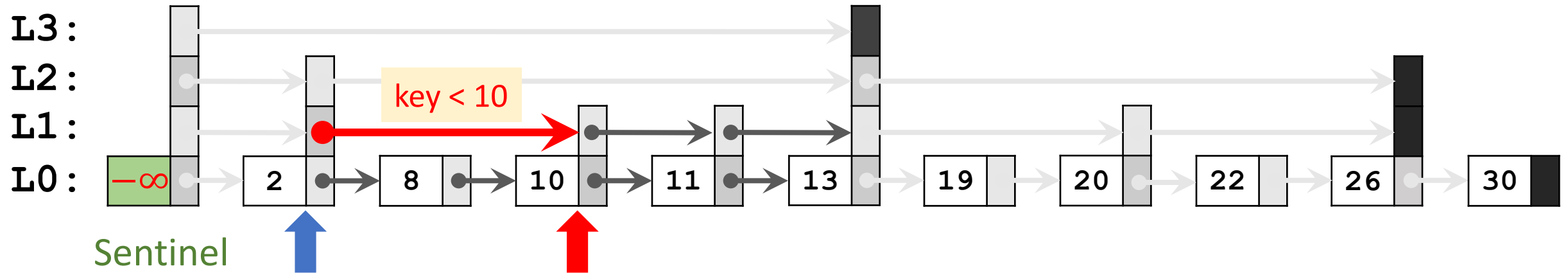
Search: **key=8**



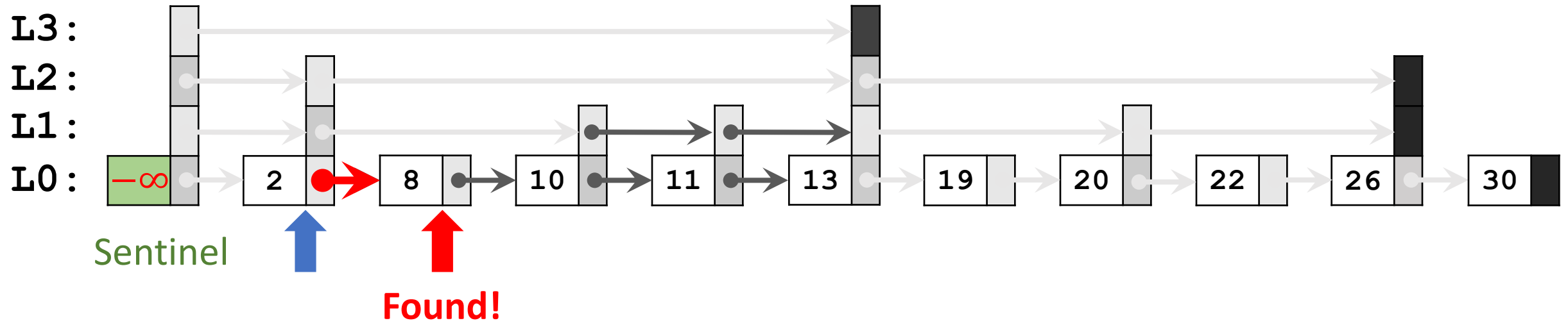
Search: **key=8**



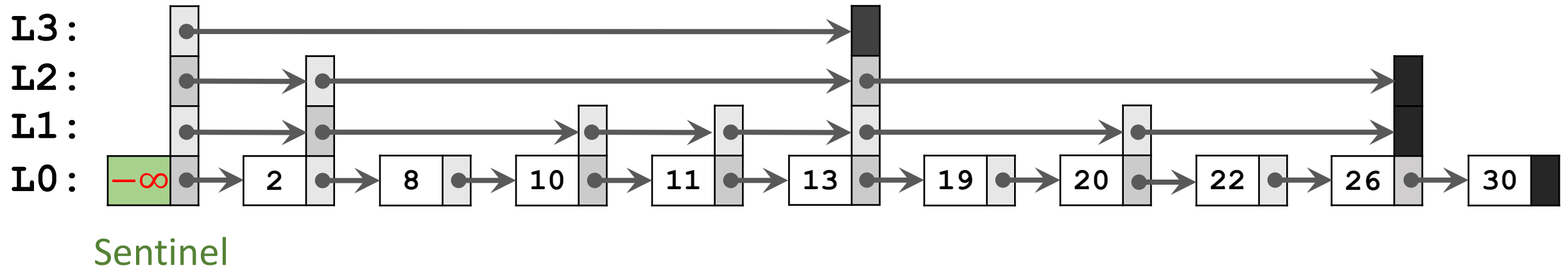
Search: **key=8**



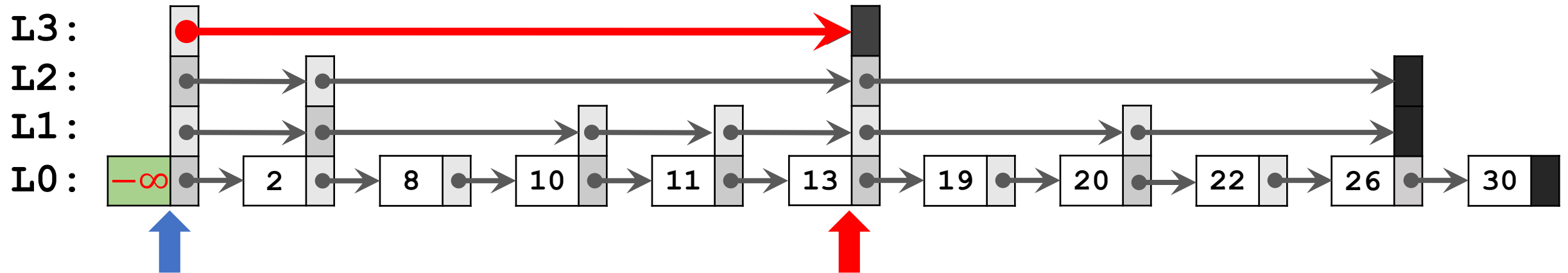
Search: **key=8**



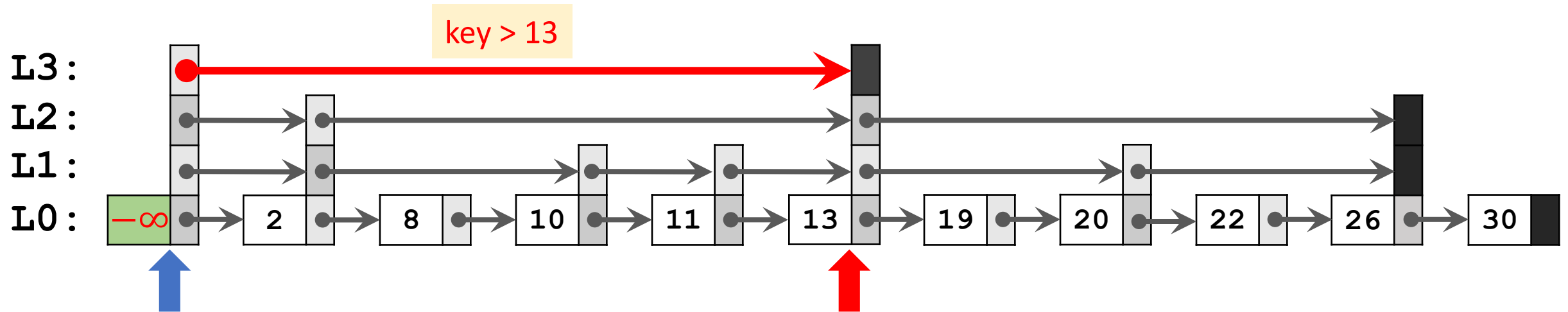
Search: **key=20**



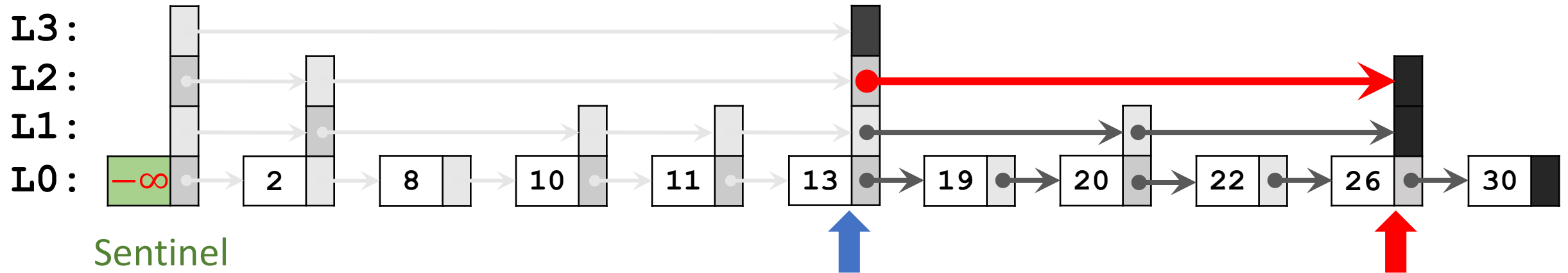
Search: **key=20**



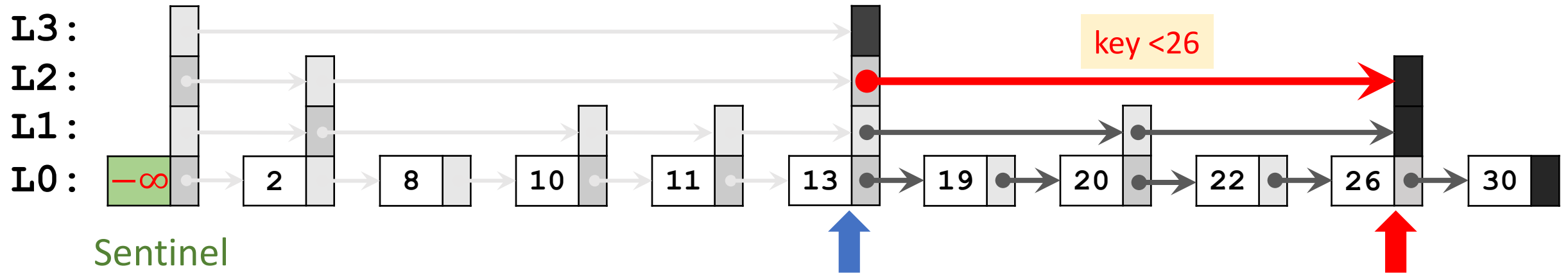
Search: **key=20**



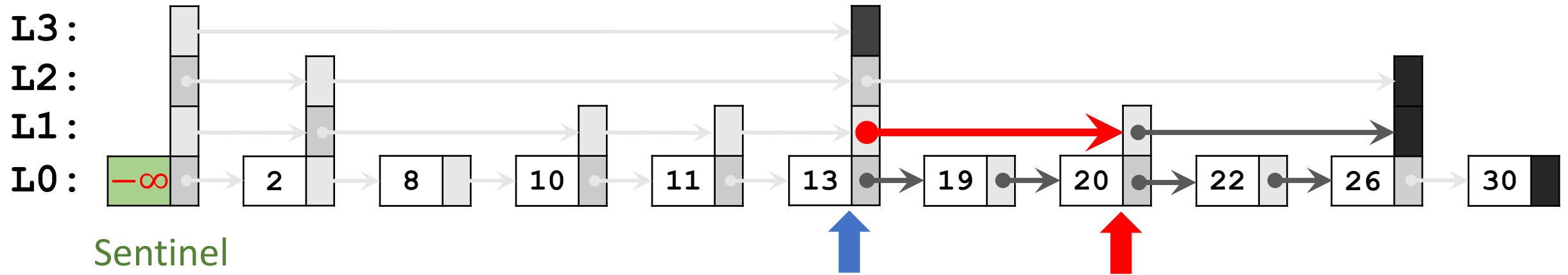
Search: **key=20**



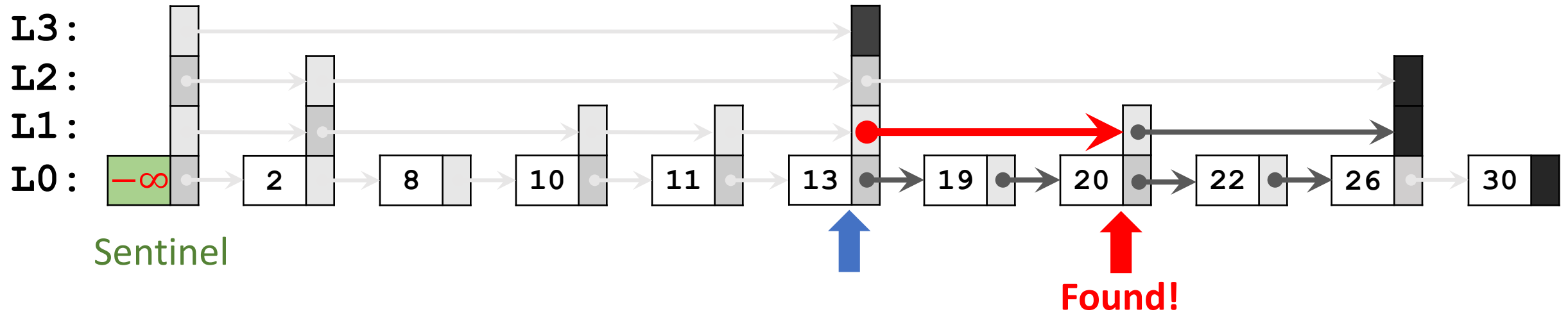
Search: **key=20**



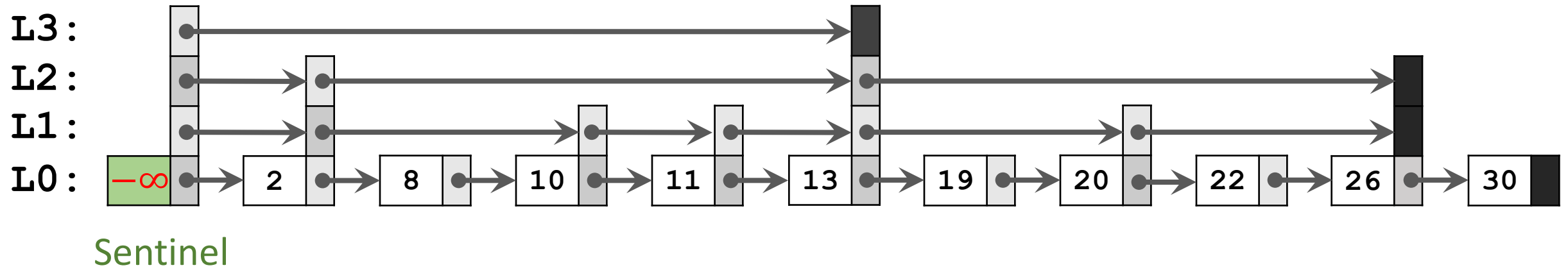
Search: **key=20**



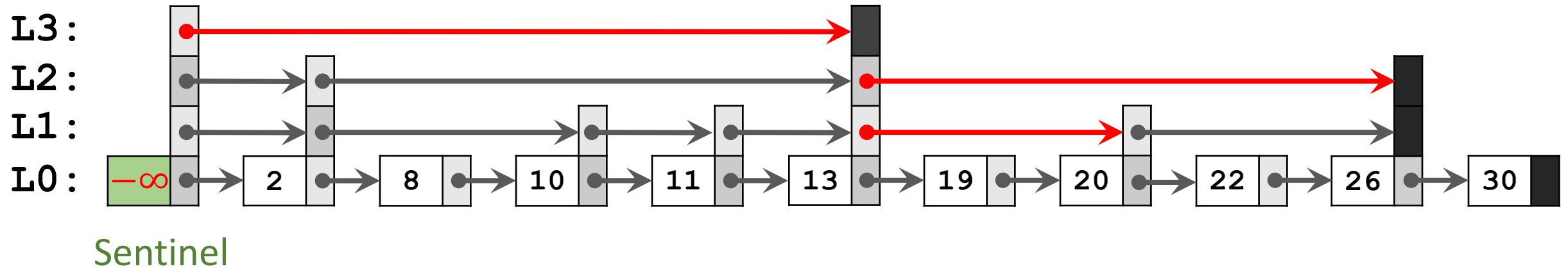
Search: **key=20**



Search: **key=21**

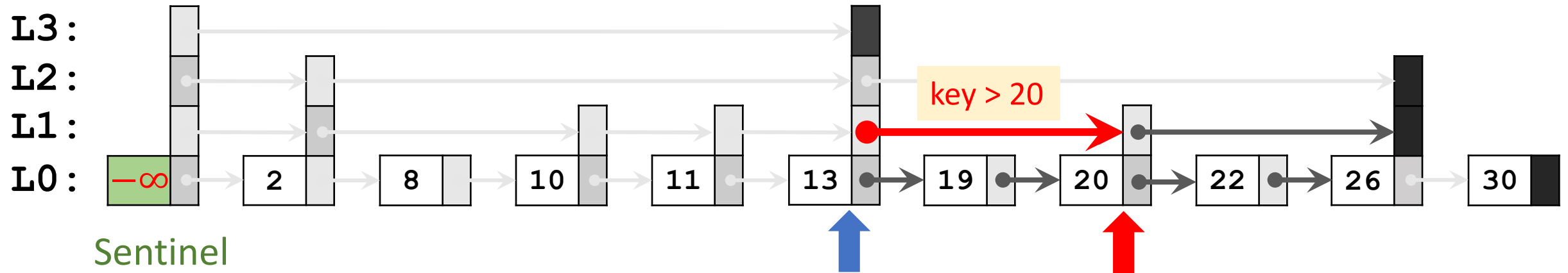


Search: **key=21**

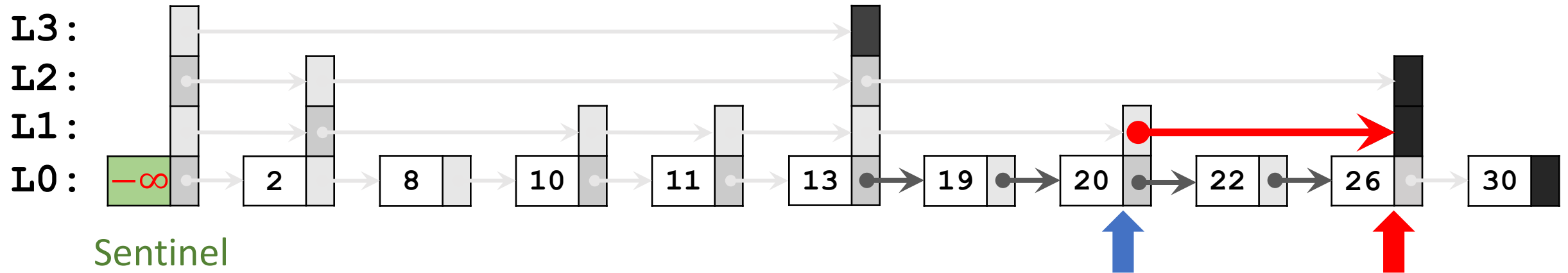


Do the same as before: traverse L3, L2, and then L1.

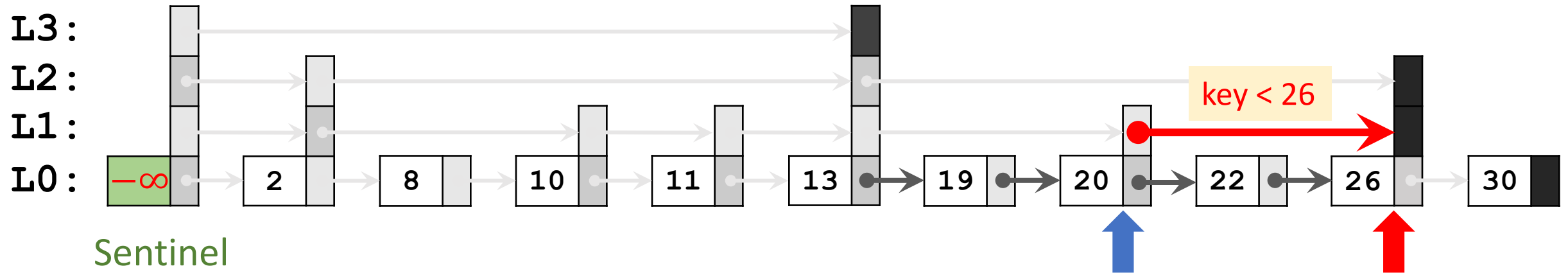
Search: **key=21**



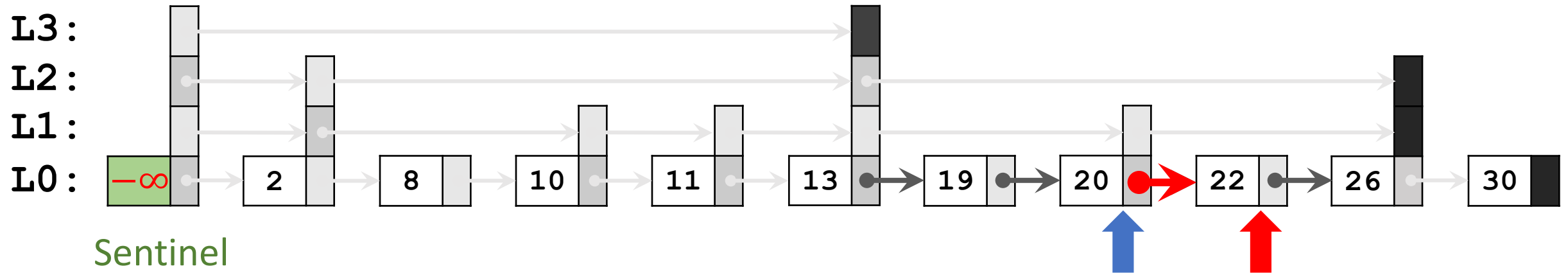
Search: **key=21**



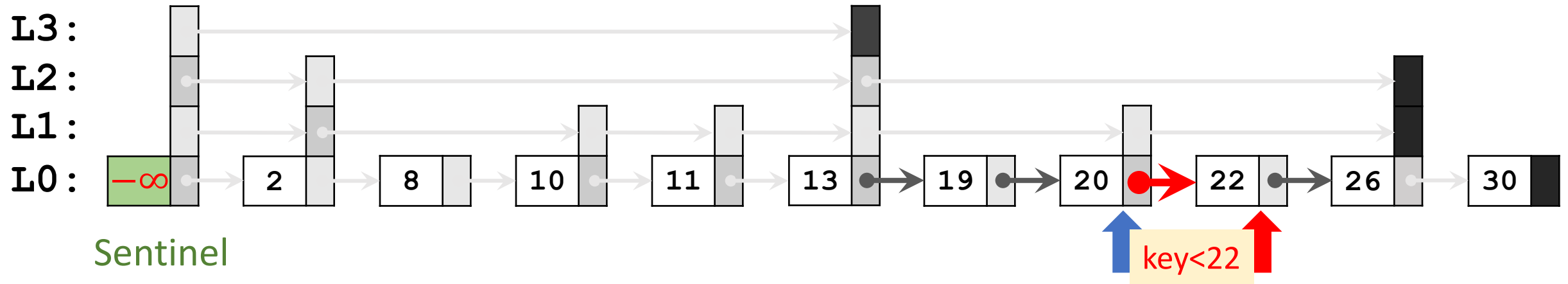
Search: **key=21**



Search: **key=21**



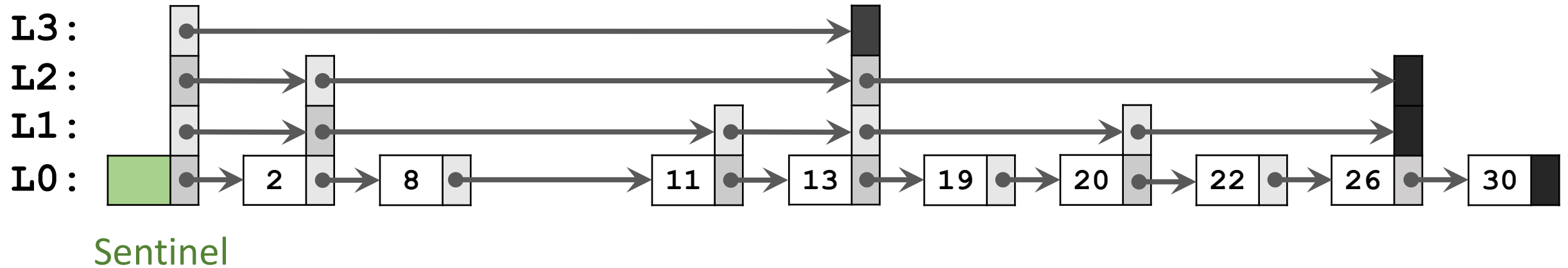
Search: **key=21**



Insertion

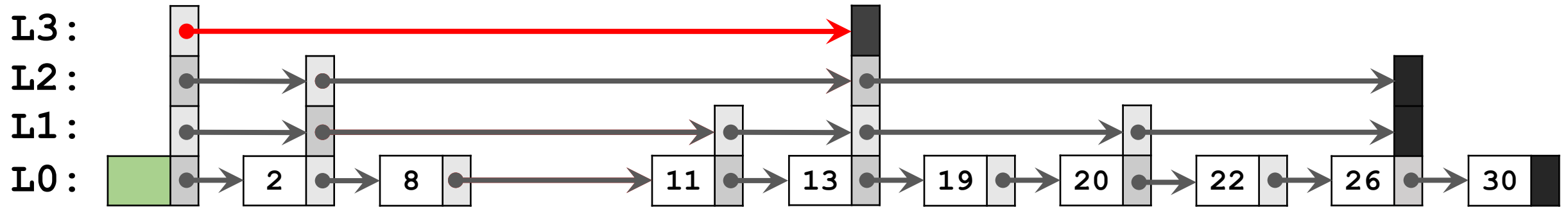
Insert (**key=9**)

First, search **key=9** and record the path.



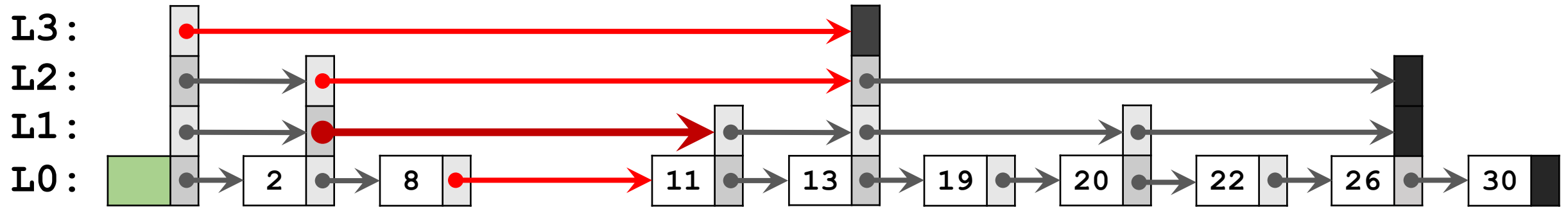
Insert (**key=9**)

First, search **key=9** and record the path.



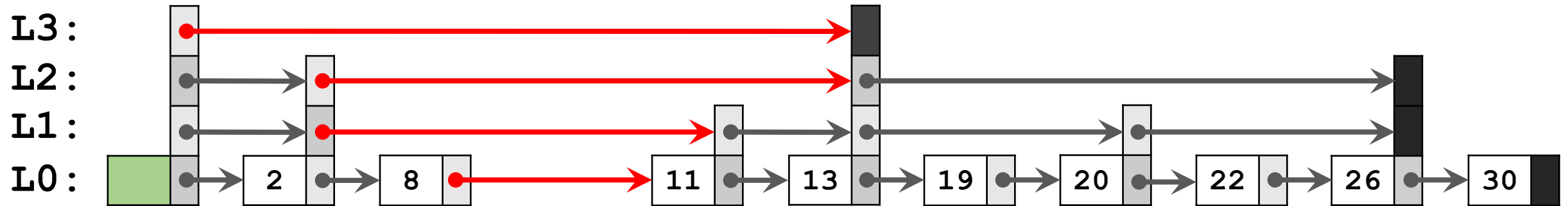
Insert (**key=9**)

First, search **key=9** and record the path.



Insert (**key=9**)

First, search **key=9** and record the path.

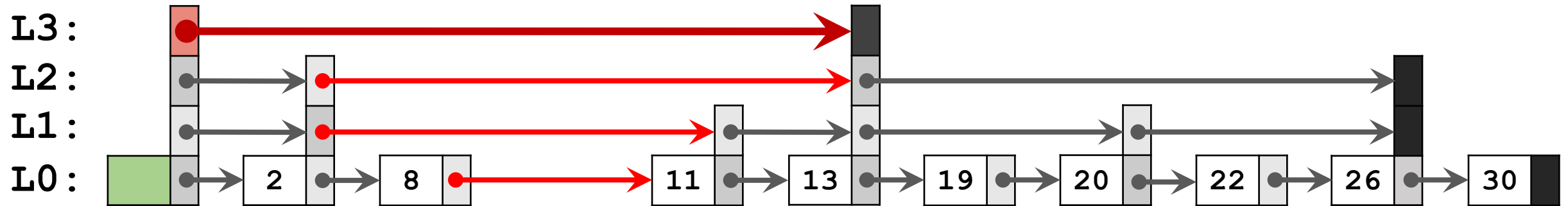


Start from:

L3:	
L2:	
L1:	
L0:	

Insert (**key=9**)

First, search **key=9** and record the path.

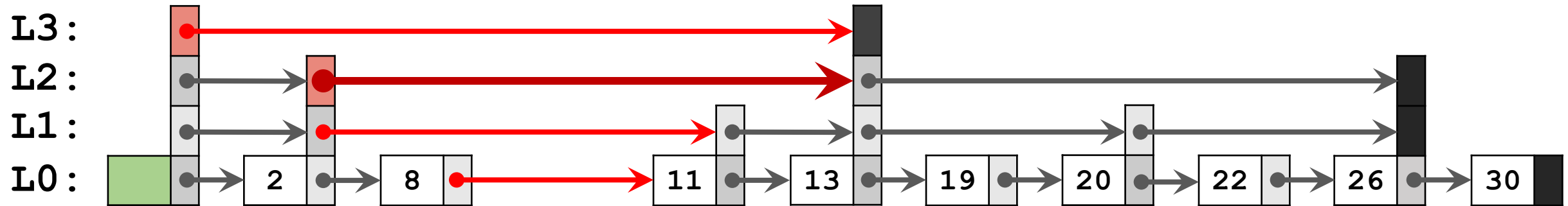


Start from:

L3 :	Sentinel
L2 :	
L1 :	
L0 :	

Insert (**key=9**)

First, search **key=9** and record the path.

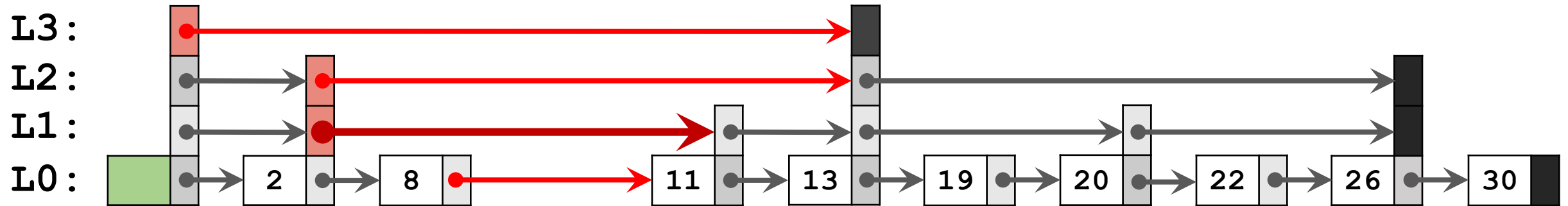


Start from:

L3:	Sentinel
L2:	Node 2
L1:	
L0:	

Insert (**key=9**)

First, search **key=9** and record the path.

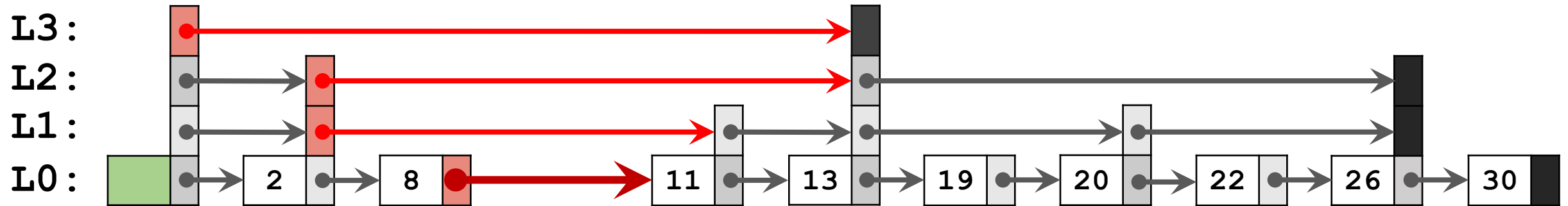


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	

Insert (**key=9**)

First, search **key=9** and record the path.

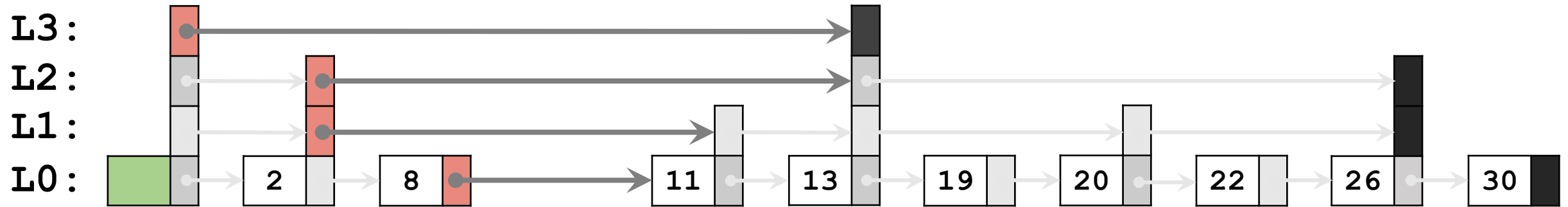


Start from:

L3 :	Sentinel
L2 :	Node 2
L1 :	Node 2
L0 :	Node 8

Insert (**key=9**)

Second, create a node whose height is random, e.g., height=2.



Create a node:

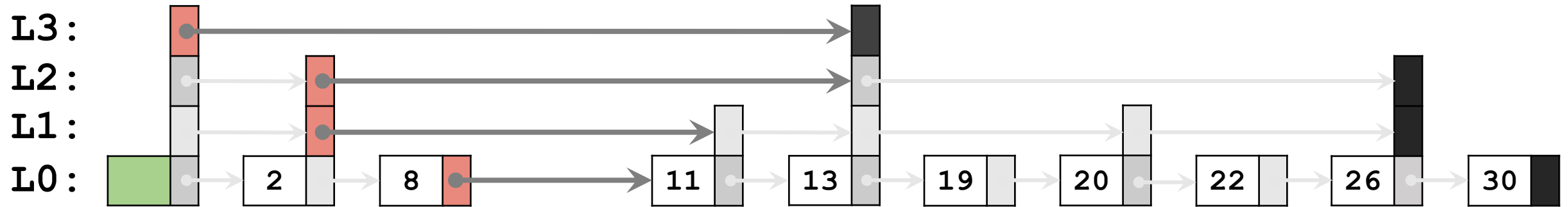


Start from:

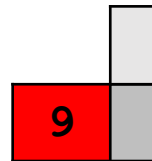
L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Second, create a node whose height is random, e.g., height=2.



Grow it height?

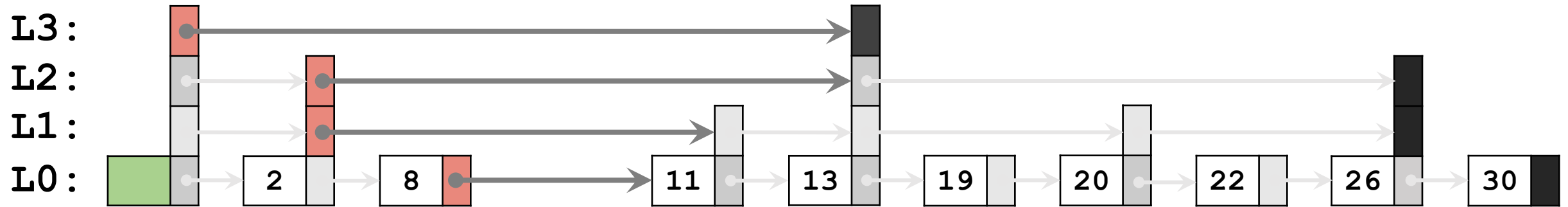


Start from:

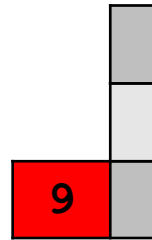
L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Second, create a node whose height is random, e.g., height=2.



Grow it height?

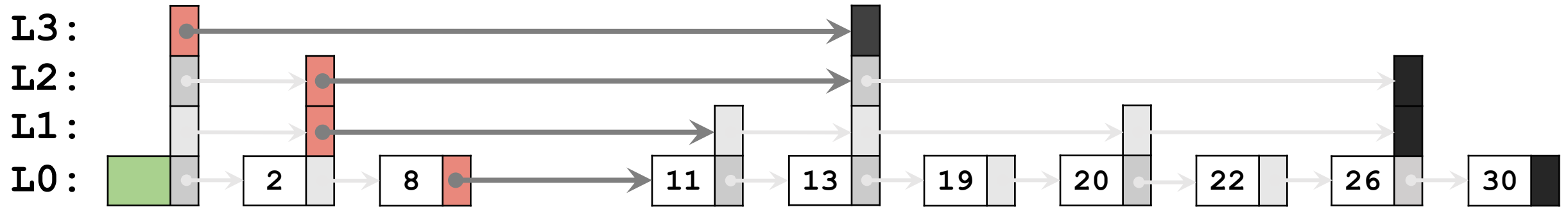


Start from:

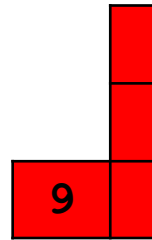
L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Second, create a node whose height is random, e.g., height=2.



New Node
(height=2) :

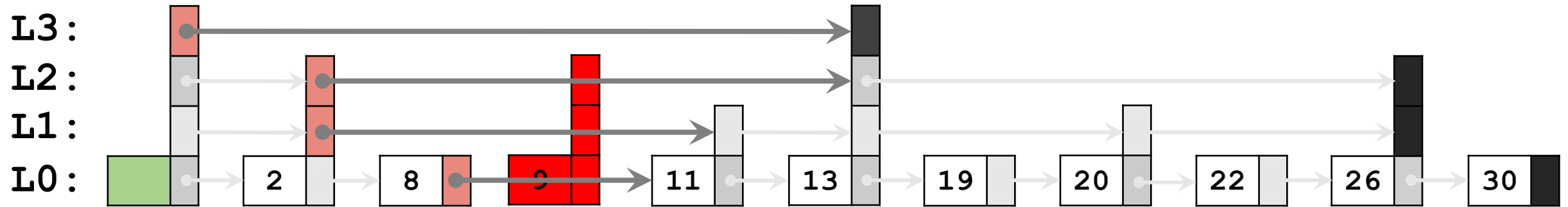


Start from:

L3 :	Sentinel
L2 :	Node 2
L1 :	Node 2
L0 :	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

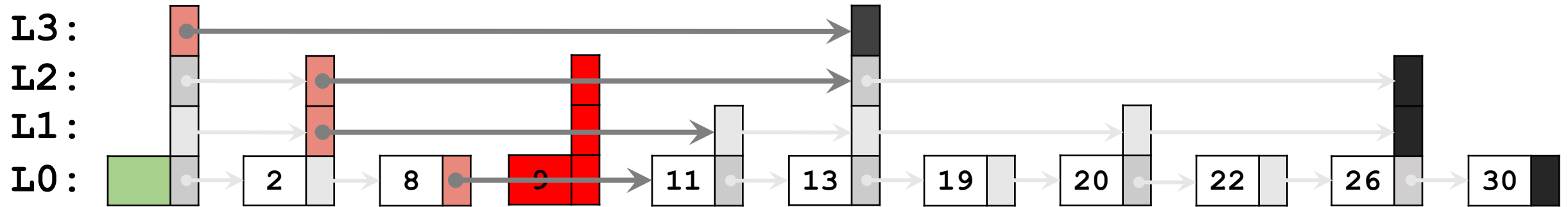


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

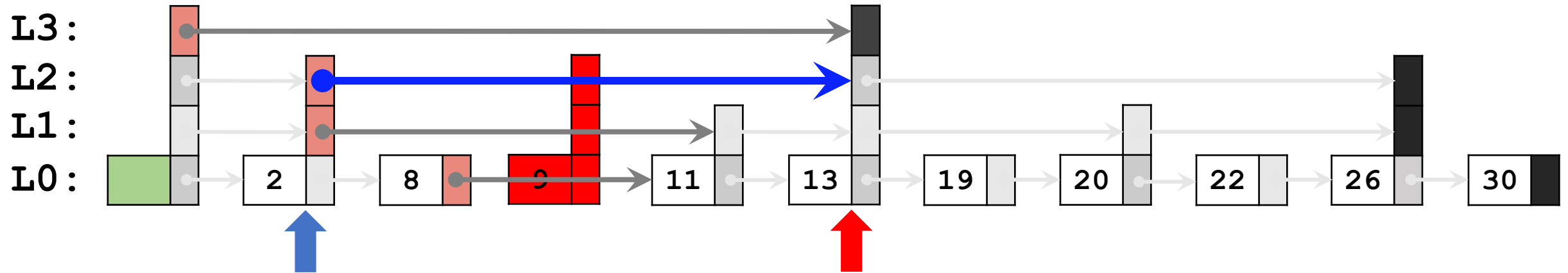


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

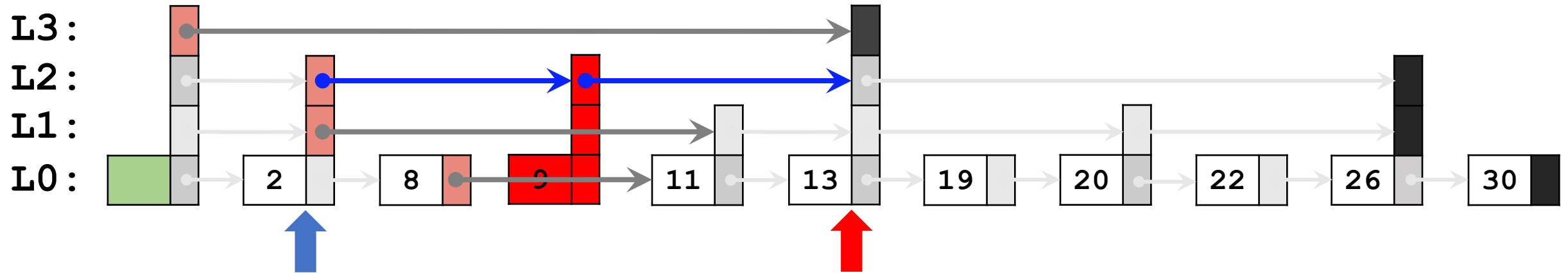


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

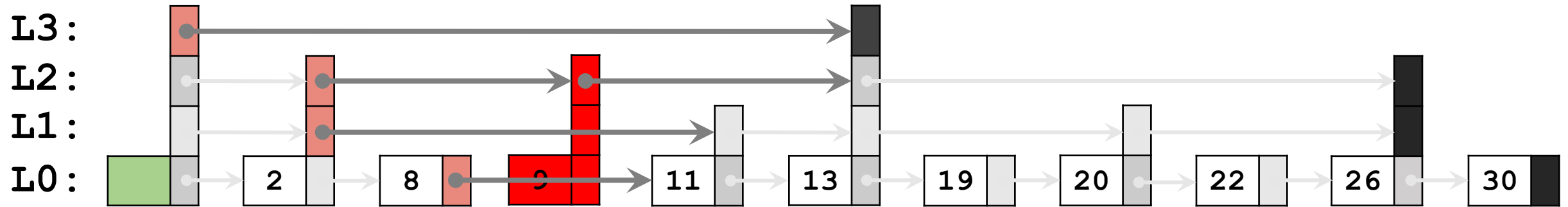


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

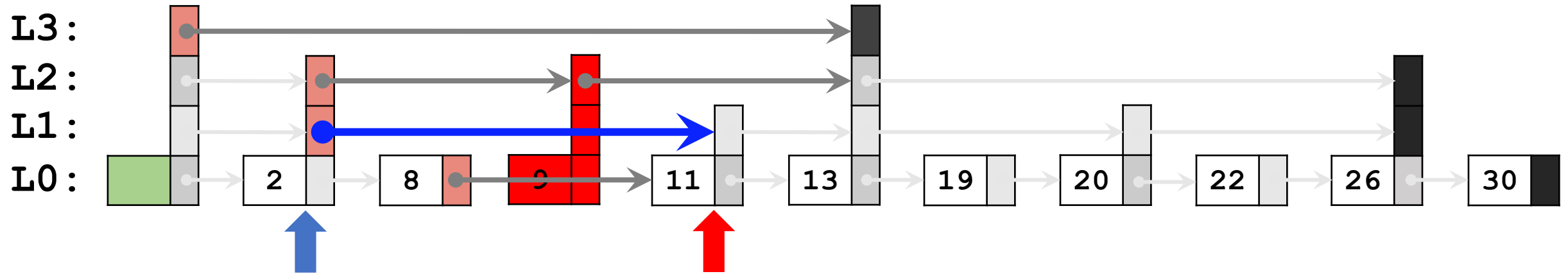


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

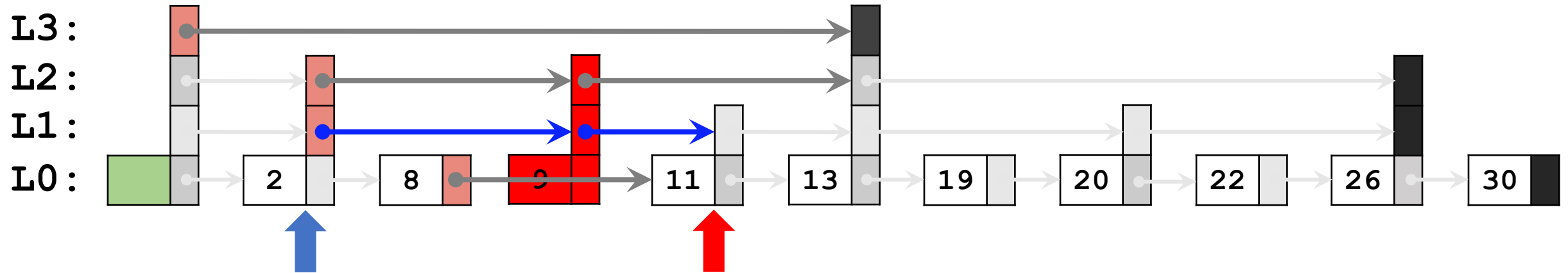


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

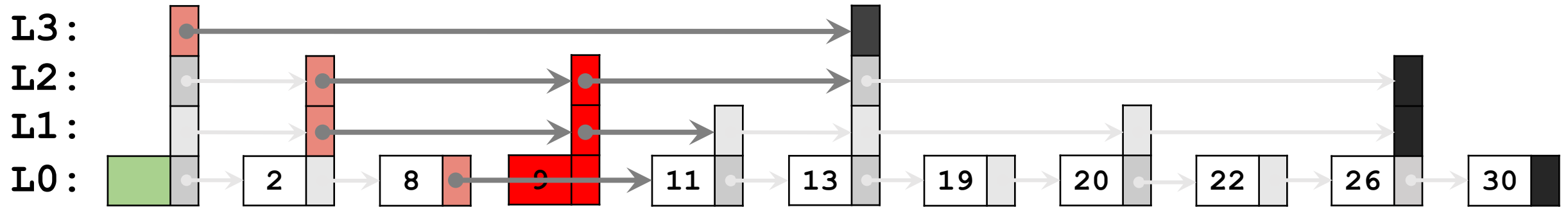


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

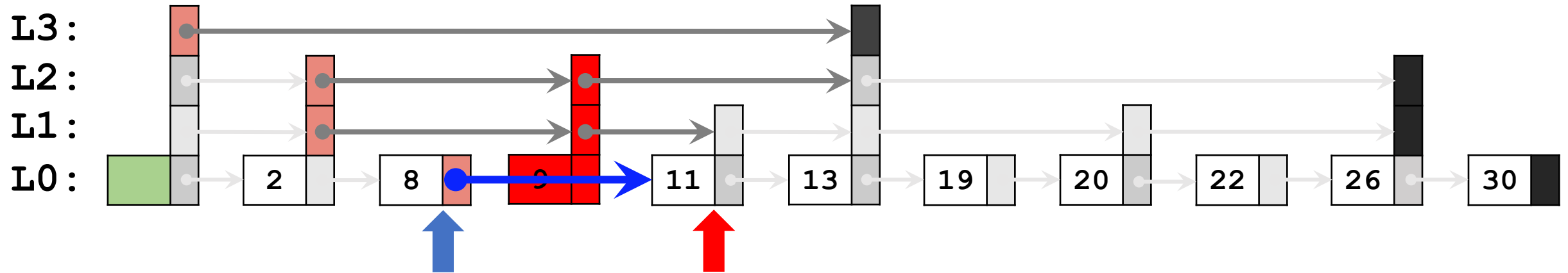


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (**key=9**)

Third, link the new node to the skip list.

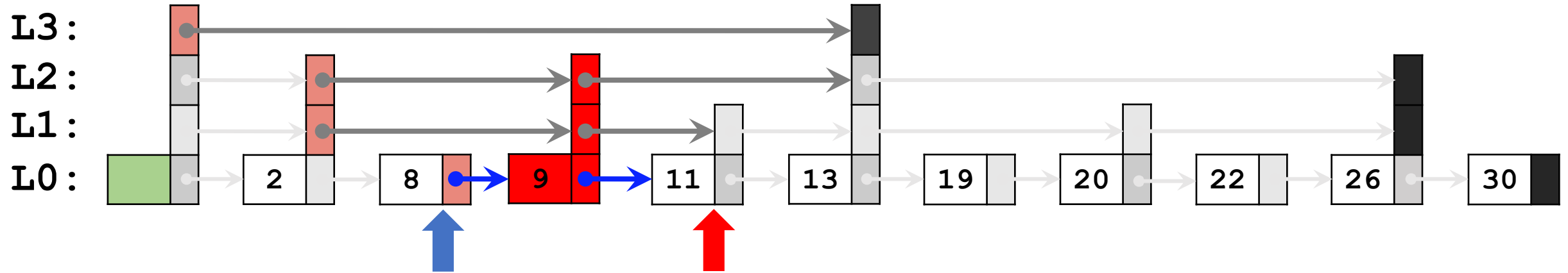


Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Insert (key=9)

Third, link the new node to the skip list.



Start from:

L3:	Sentinel
L2:	Node 2
L1:	Node 2
L0:	Node 8

Thank You!

<http://wangshusen.github.io/>