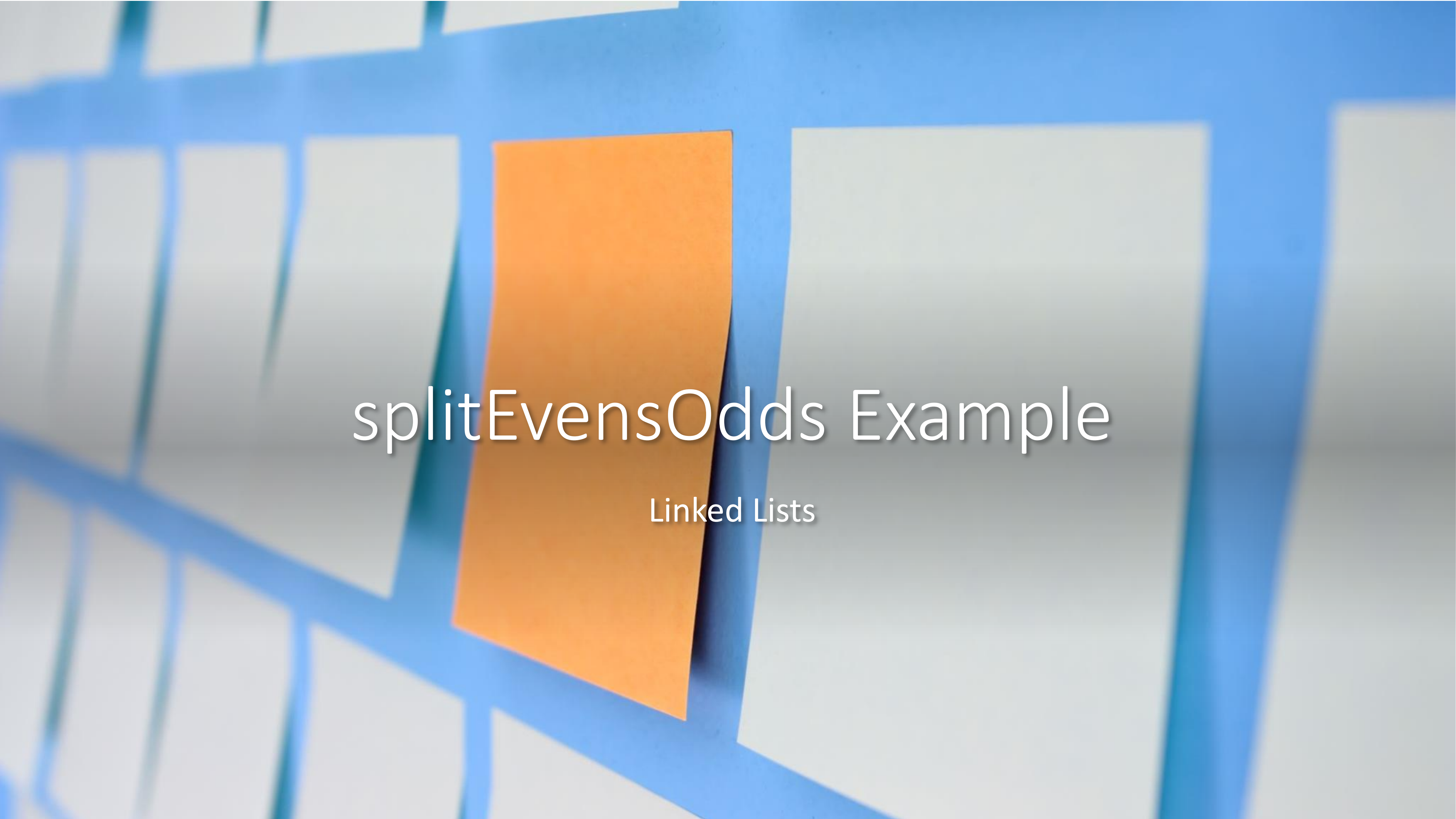


splitEvensOdds Example

Linked Lists

Algorithm for splitEvensOdds

Assume `list` is the original list and the `evensList` and `oddsList` are passed as references to the function:

```
void
splitEvensOdds(intLinkedList &evens, intLinkedList
&odds);
```

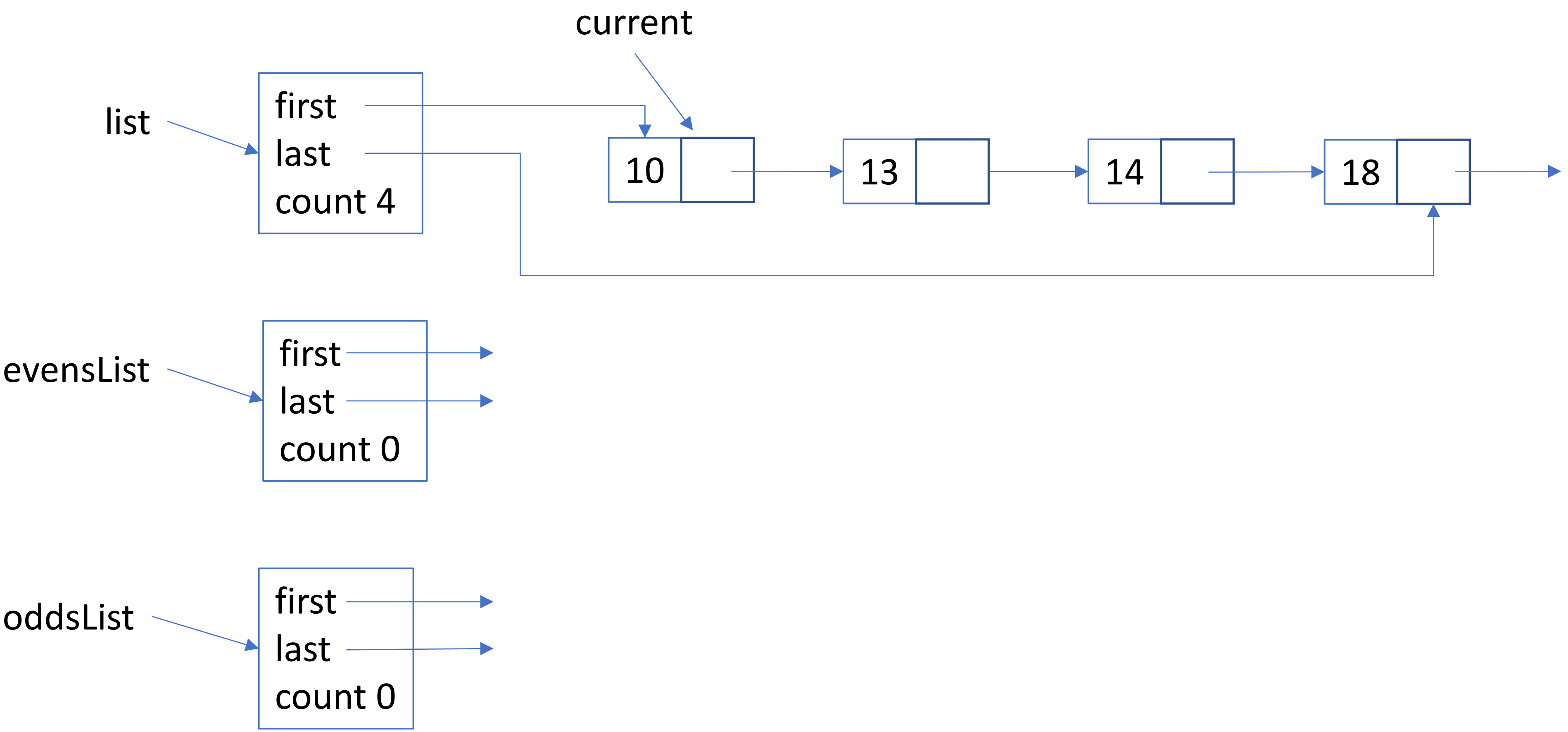
In client:

```
list.splitEvensOdds(evens,
odds);
```

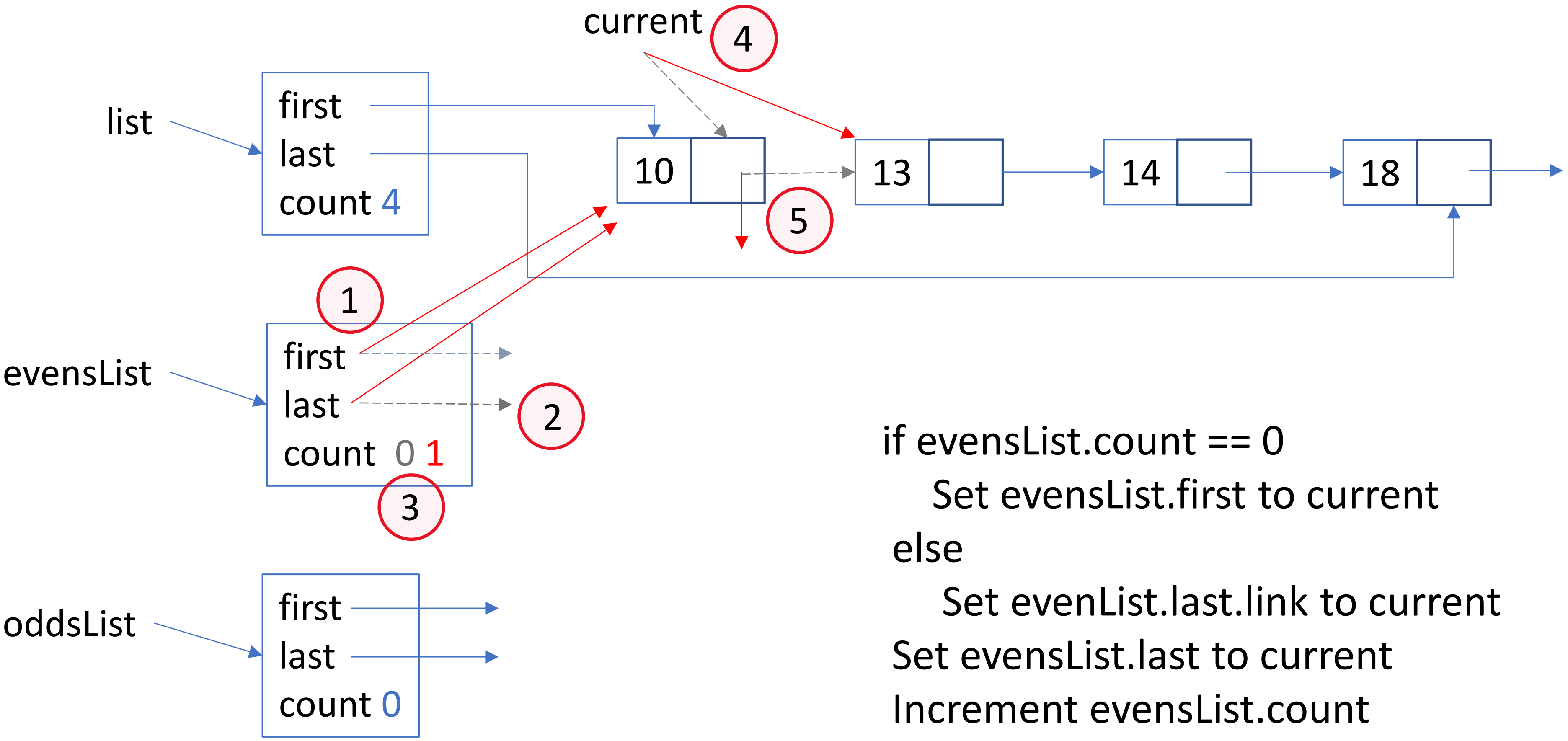
```
Declare current as a pointer to an intLinkedList node
Set current to list.first
While current is not nullptr
  If cursor.info is even
     if evensList.count == 0
         Set evensList.first to current
     else
         Set evenList.last.link to current
     Set evensList.last to current
     Increment evensList.count
     Set current to current.link
     Set  evensList.last.link to nullptr
  else
     Do the same steps for oddsList

// Clean up so list is an empty list:
Set list.first to nullptr
Set list.last to null ptr
Set list.count to 0
```

Initial Conditions



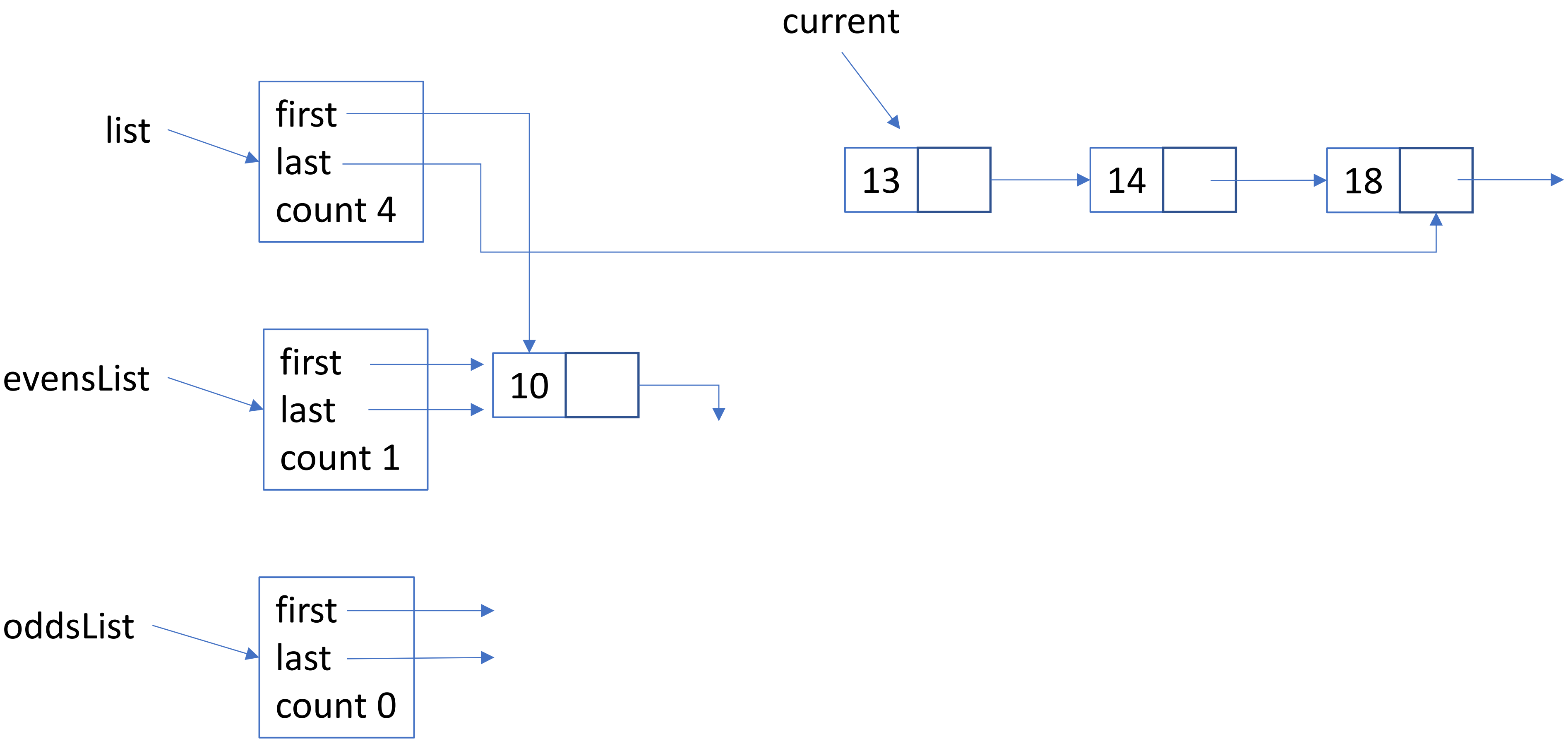
Adding node with 10 to evensList



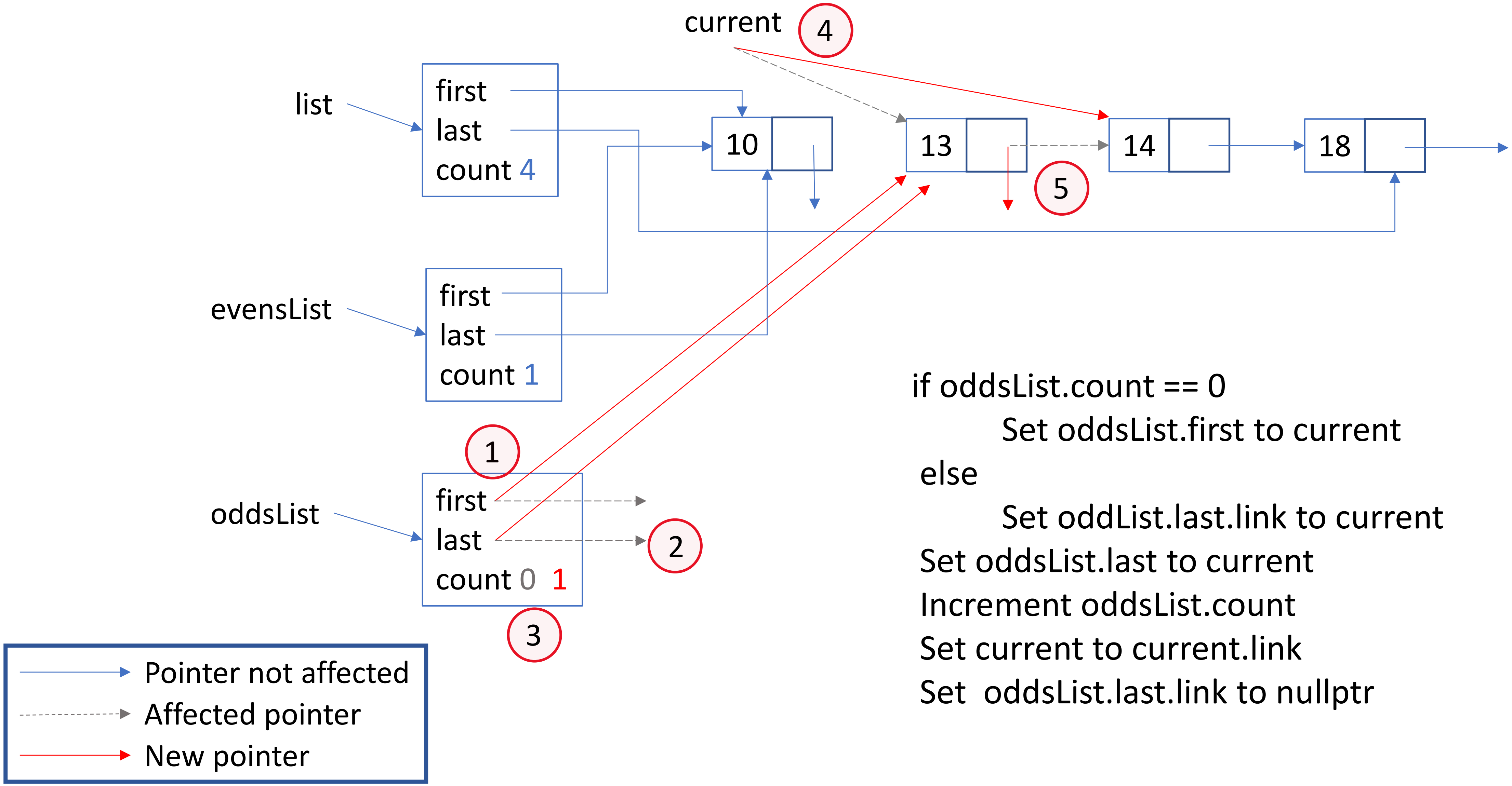
```
if evensList.count == 0
    Set evensList.first to current
else
    Set evenList.last.link to current
Set evensList.last to current
Increment evensList.count
Set current to current.link
Set  evensList.last.link to nullptr
```

→ Pointer not affected
⇢ Affected pointer
→ New pointer

Effect of adding node 10 to **`evensList`**

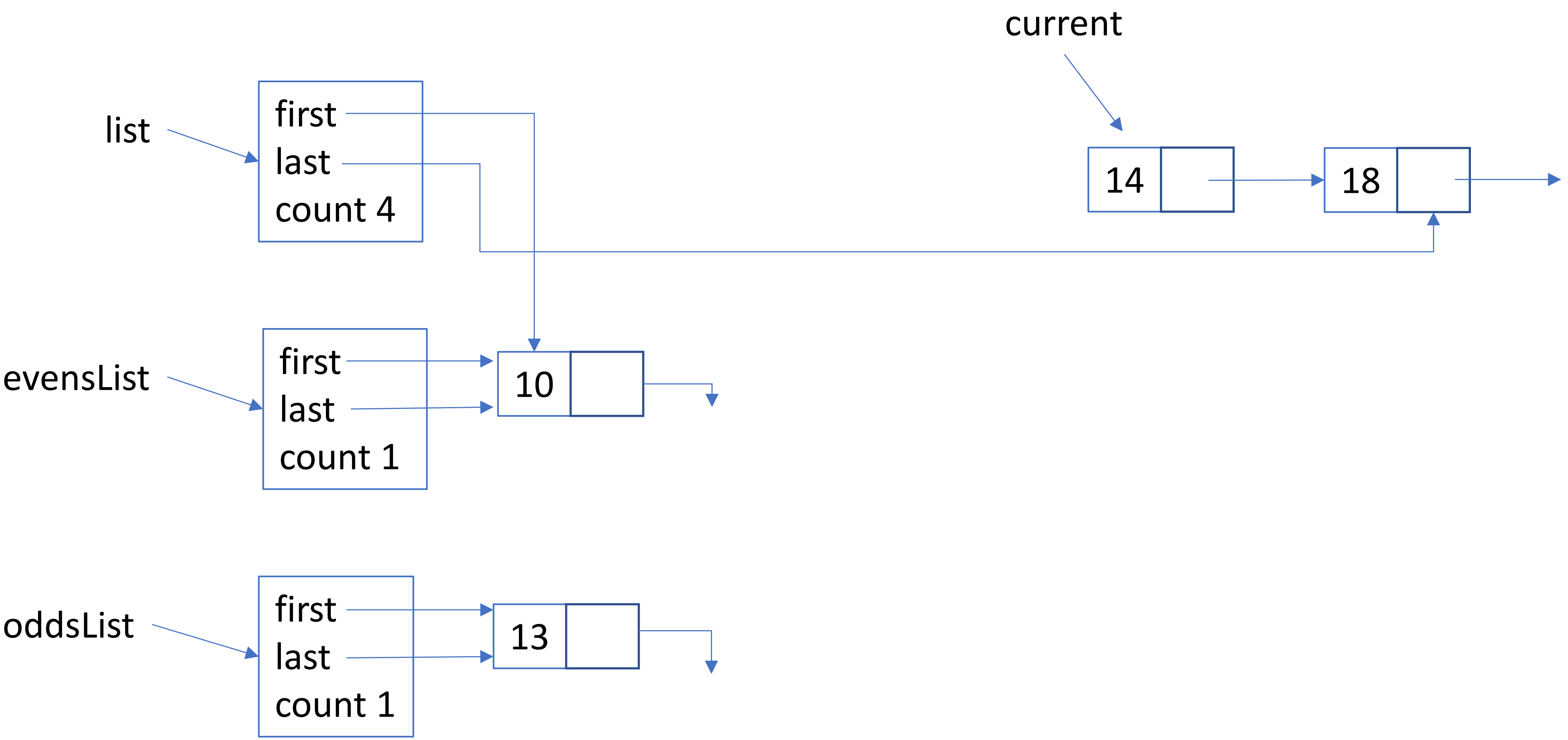


Adding node with 13 to oddsList

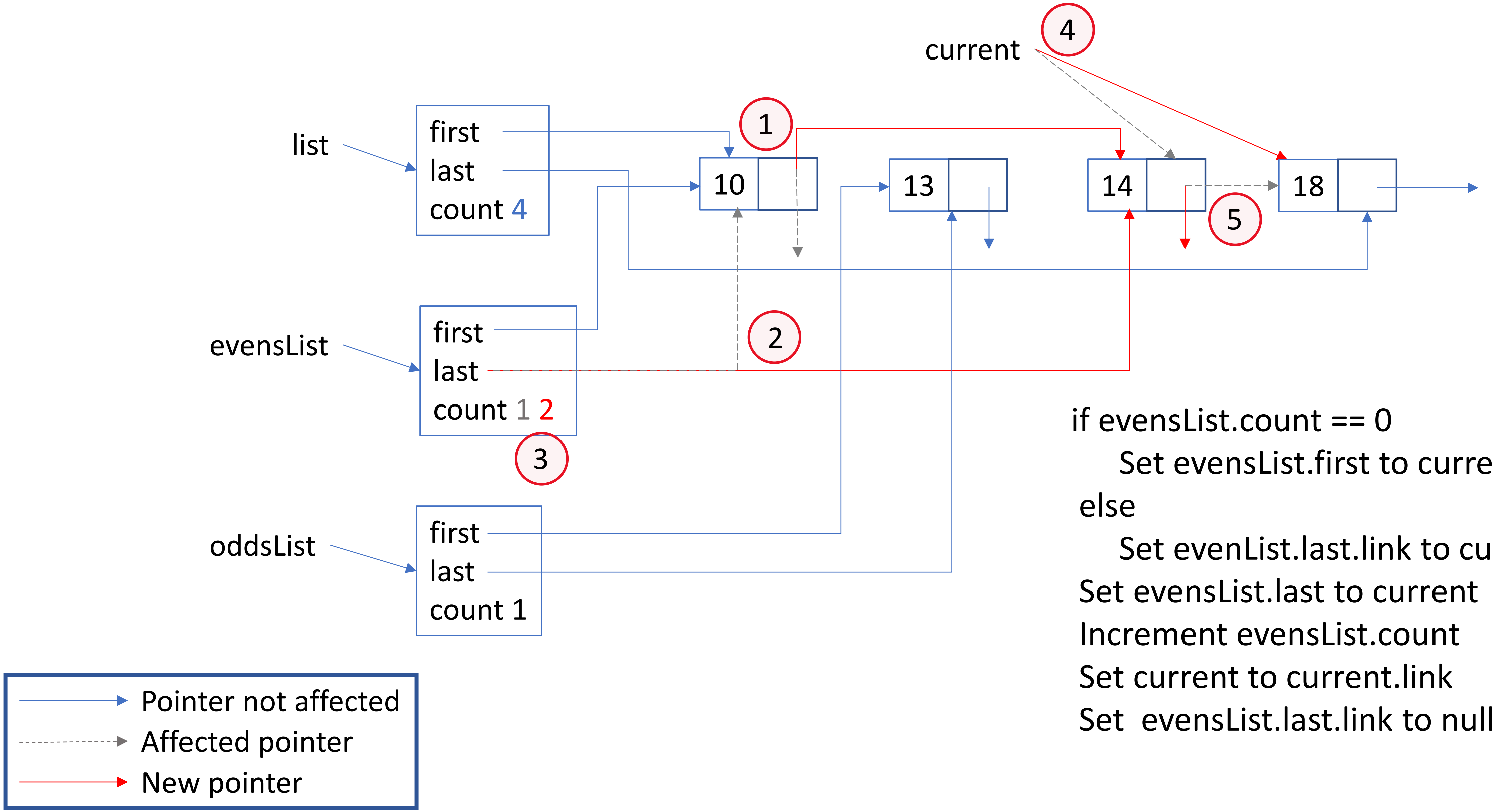


```
if oddsList.count == 0
        Set oddsList.first to current
else
        Set oddList.last.link to current
Set oddsList.last to current
Increment oddsList.count
Set current to current.link
Set  oddsList.last.link to nullptr
```

Effect of adding node 13 to **`oddsList`**

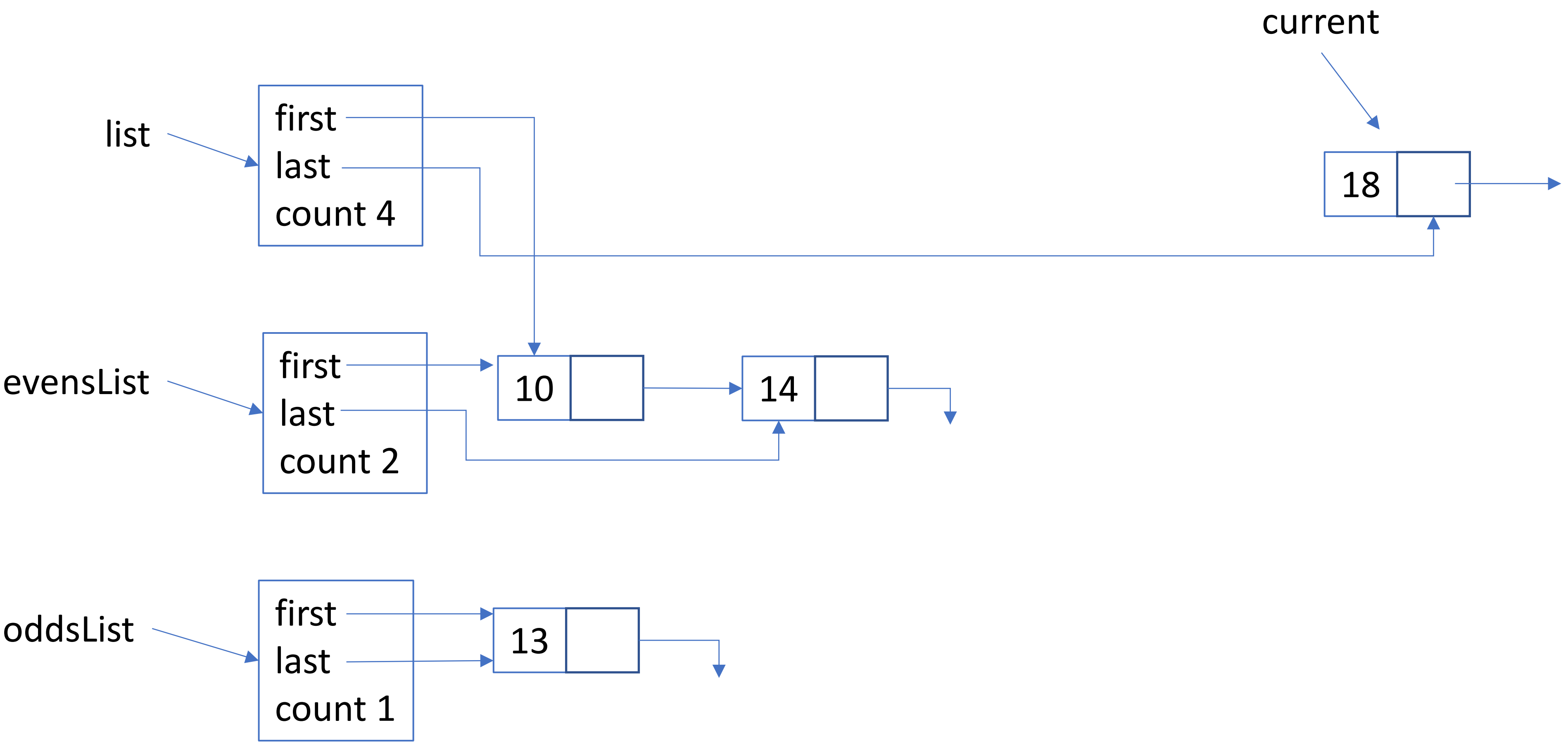


Adding node with 14 to evensList

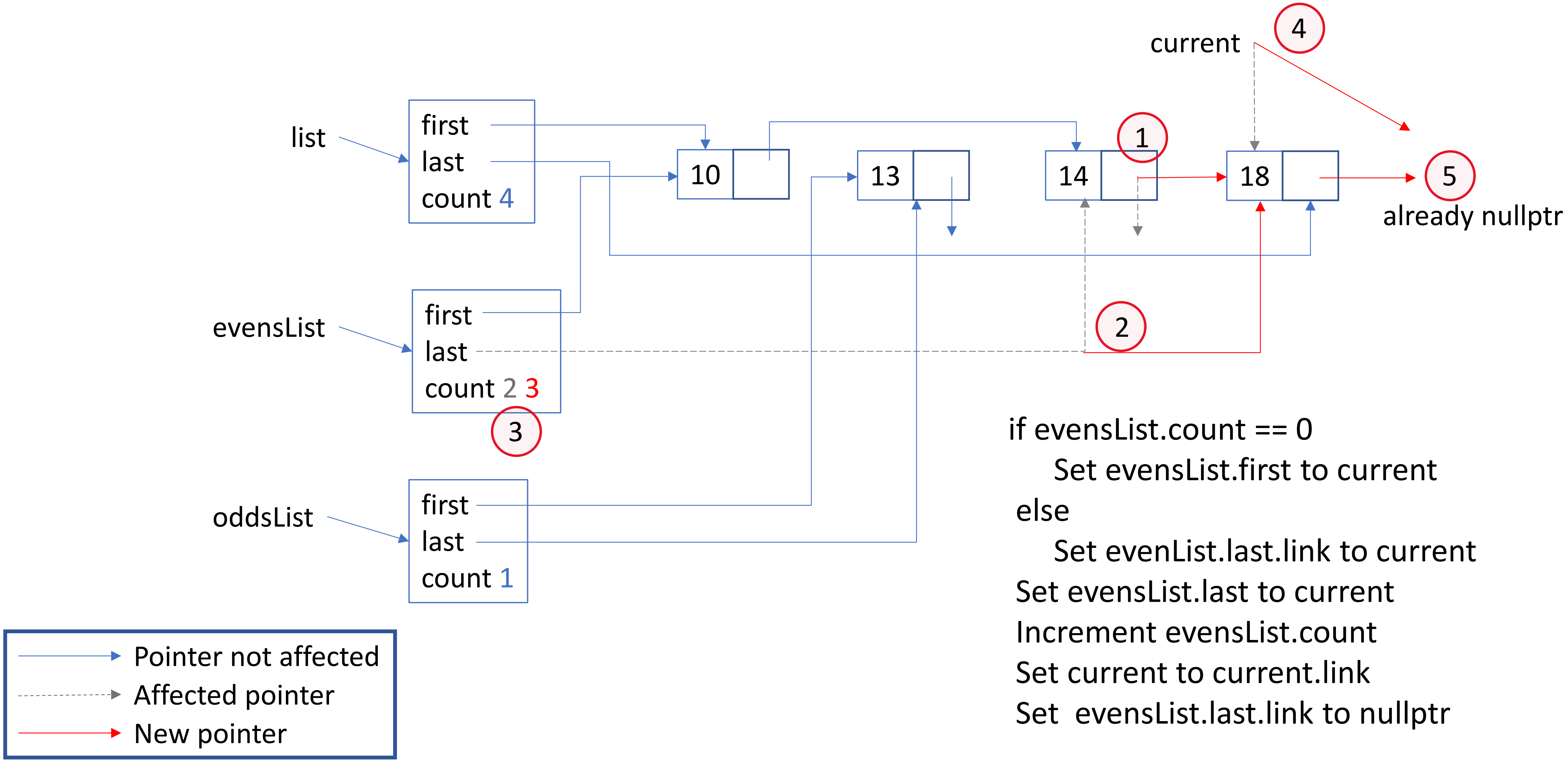


```
if evensList.count == 0
    Set evensList.first to current
else
    Set evenList.last.link to current
Set evensList.last to current
Increment evensList.count
Set current to current.link
Set  evensList.last.link to nullptr
```

Effect of adding node 14 to `evensList`

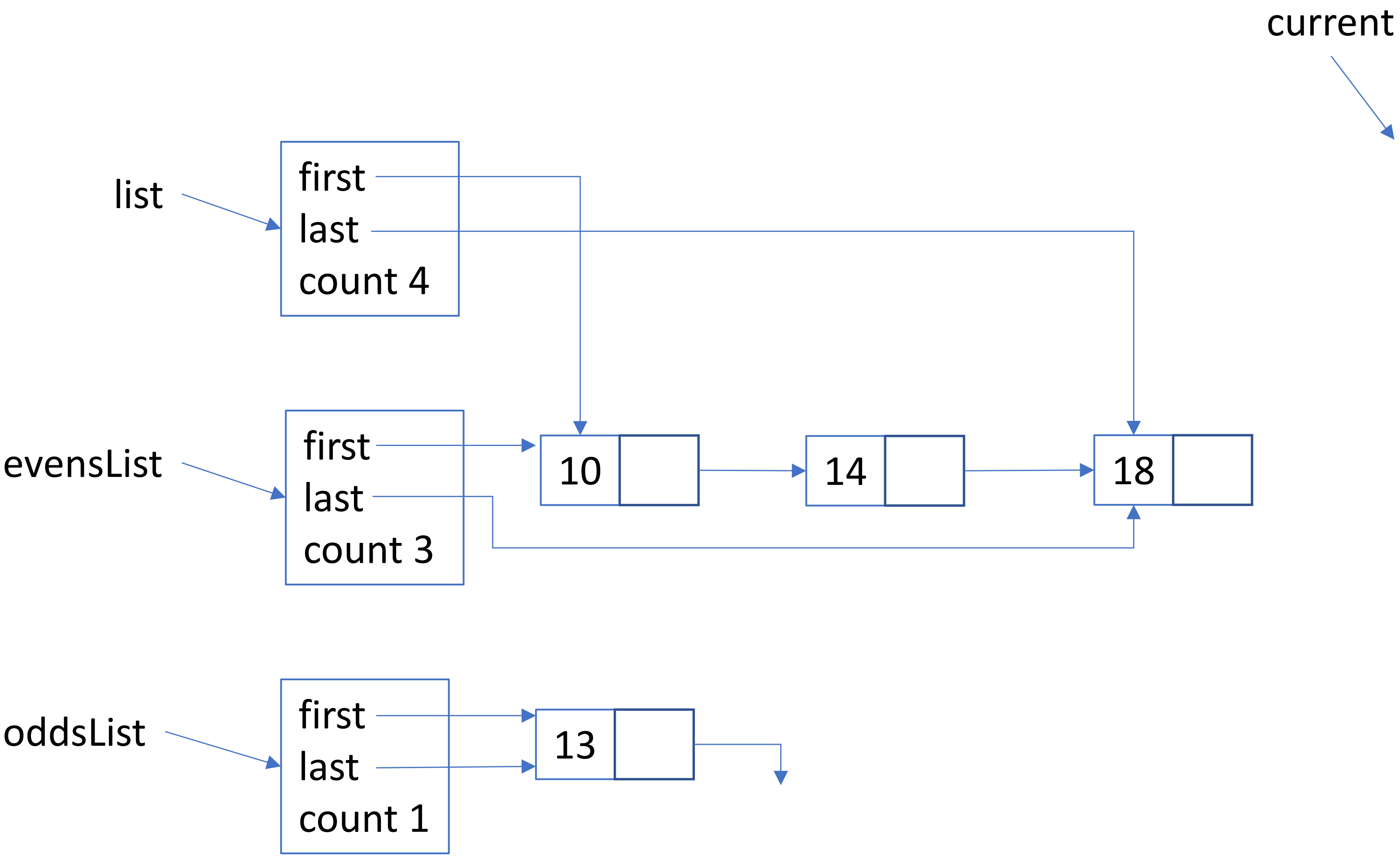


Adding node with 18 to evensList

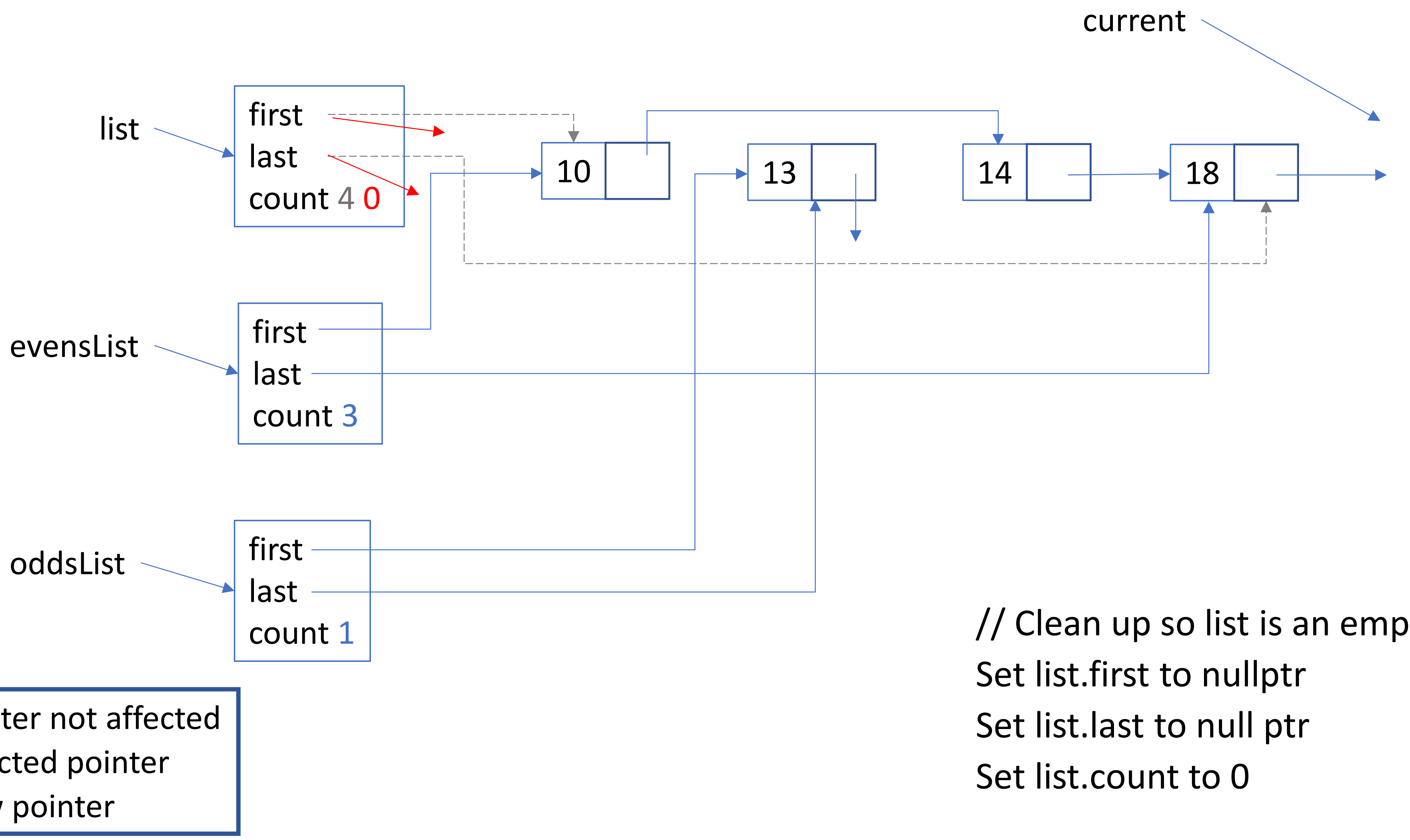


```
if evensList.count == 0
    Set evensList.first to current
else
    Set evenList.last.link to current
Set evensList.last to current
Increment evensList.count
Set current to current.link
Set  evensList.last.link to nullptr
```

Effect of adding node 18 to **evensList**



Cleaning up **`list`**



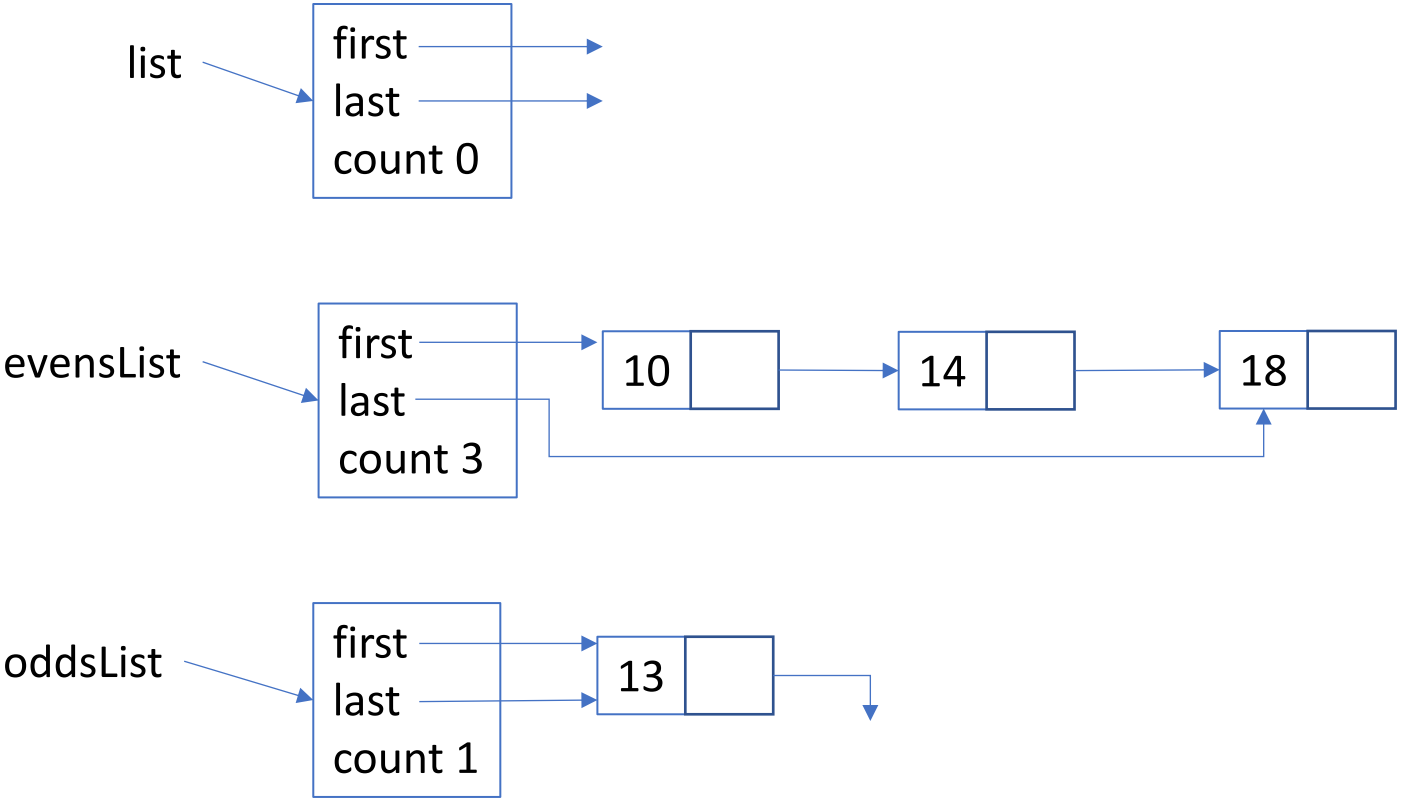
// Clean up so list is an empty list:

Set list.first to nullptr

Set list.last to null ptr

Set list.count to 0

Effect of cleaning up `list`


list
first
last
count 0
evensList
first
last
count 3
10
14
18
oddsList
first
last
count 1
13