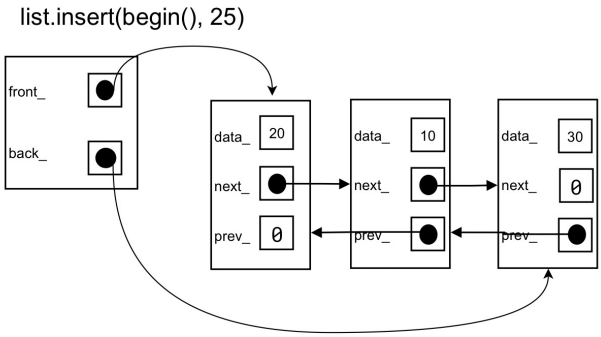
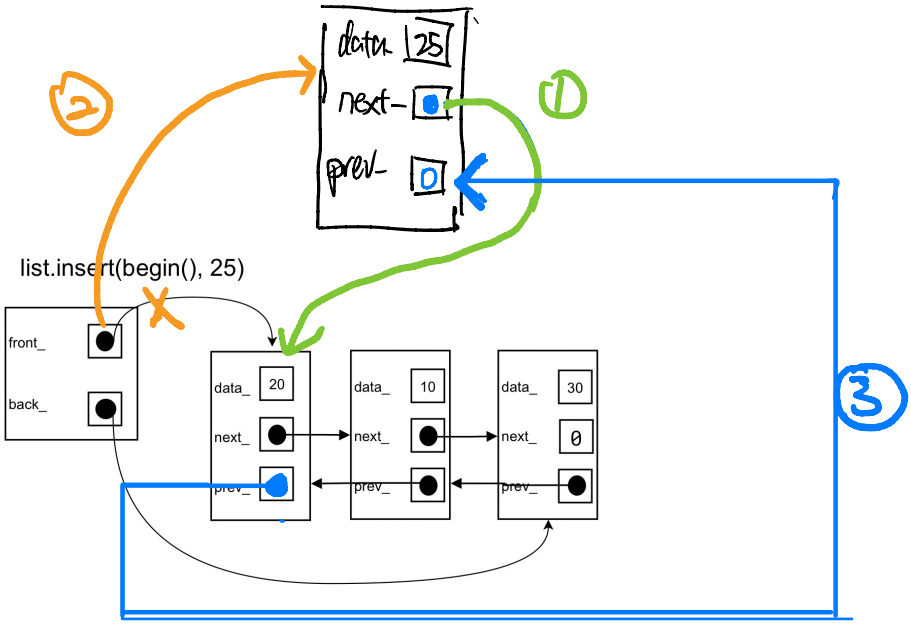
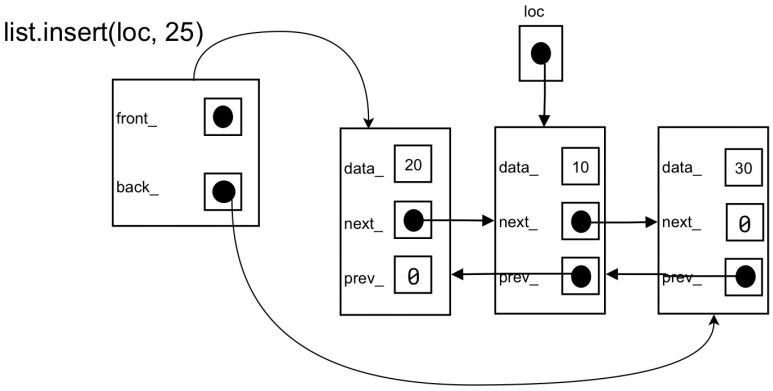
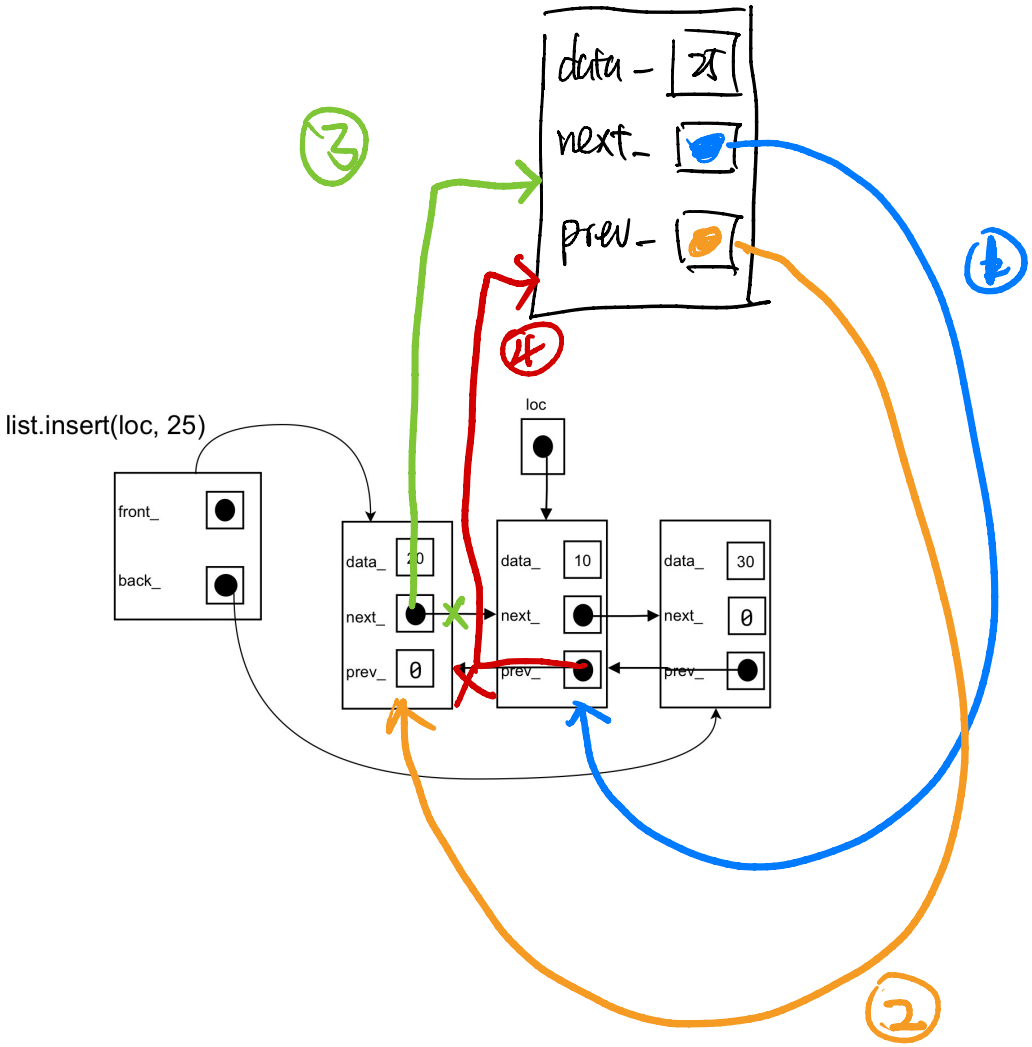
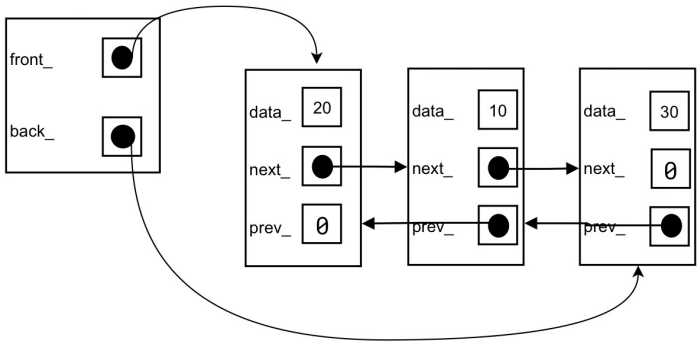

list.insert(begin(), 25)
front_
back_
data_ 20
next_
prev_ 0
data_ 10
next_
prev_
data_ 30
next_ 0
prev_


data 25
next-
prev- 0
1
2
3
list.insert(begin(), 25)
front_
back_
data_ 20
next_
data_ 10
next_
prev_
data_ 30
next_ 0
prev_

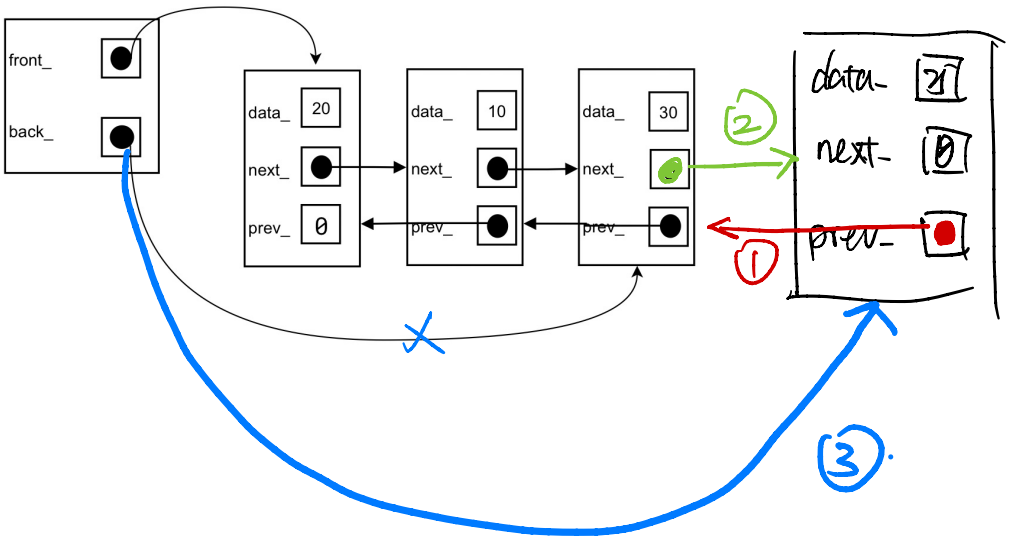

list.insert(loc, 25)
loc
front_
back_
data_ 20
next_
prev_ 0
data_ 10
next_
prev_
data_ 30
next_ 0
prev_


data_ 25
next_
prev_
3
4
1
2
list.insert(loc, 25)
loc
front_
back_
data_ 20
next_
prev_ 0
data_ 10
next_
prev_
data_ 30
next_ 0
prev_

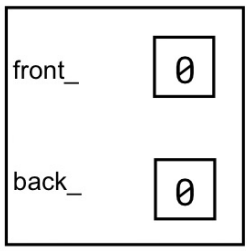
list.insert(end(), 25)



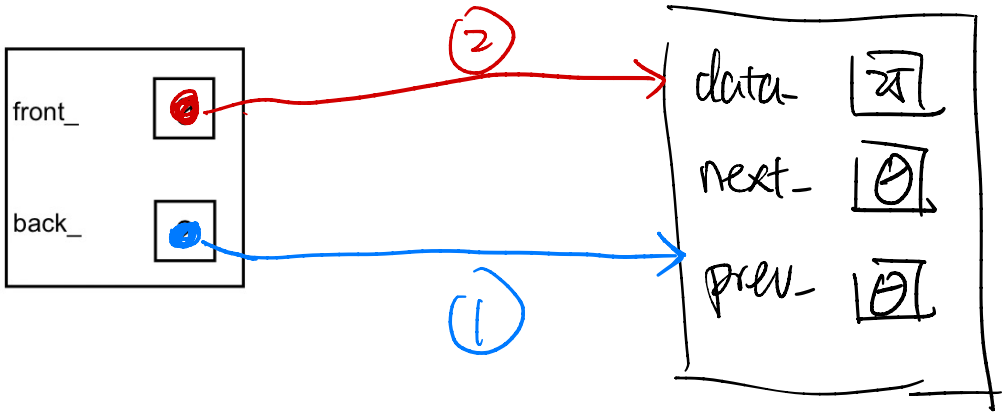
list.insert(end(), 25)



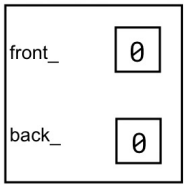
list.insert(end(), 25)



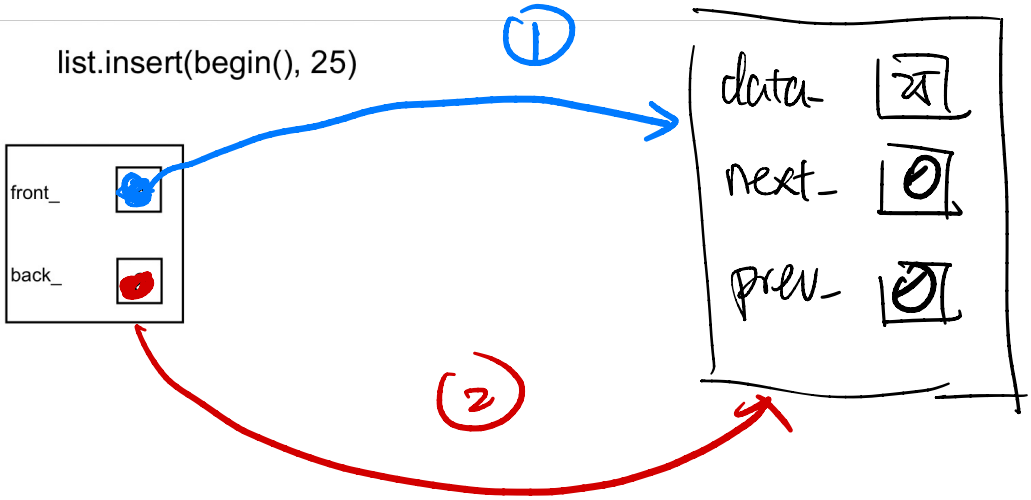
list.insert(end(), 25)



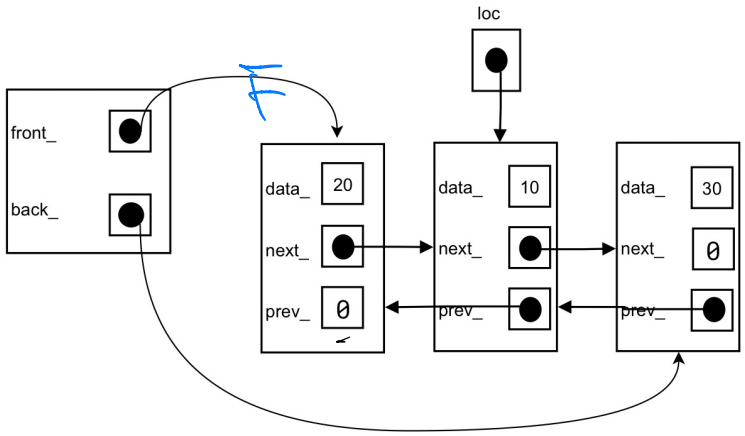
list.insert(begin(), 25)



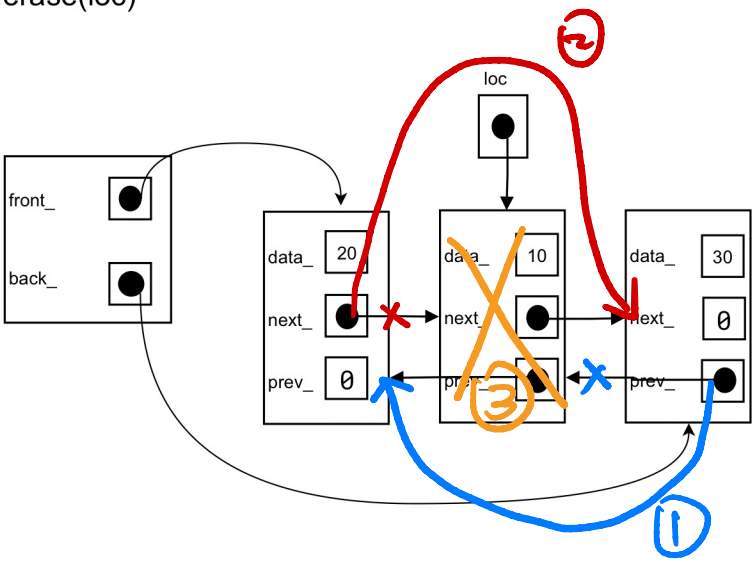
list.insert(begin(), 25)

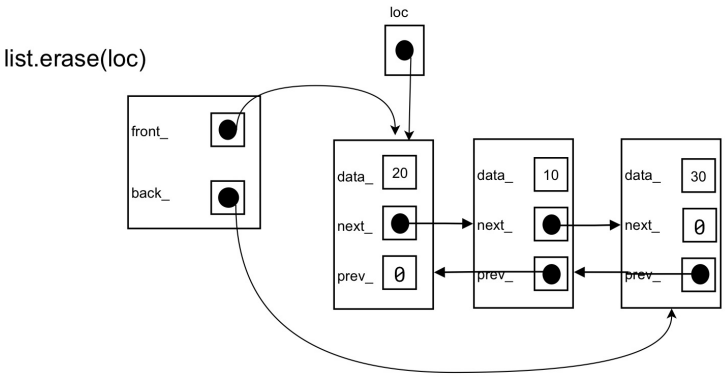


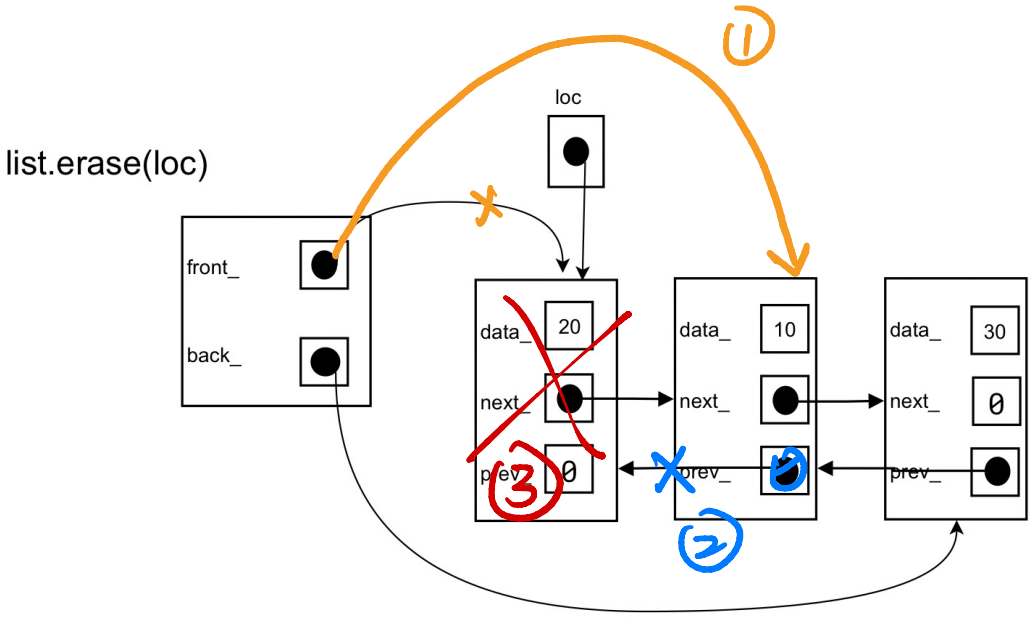
list.erase(loc)



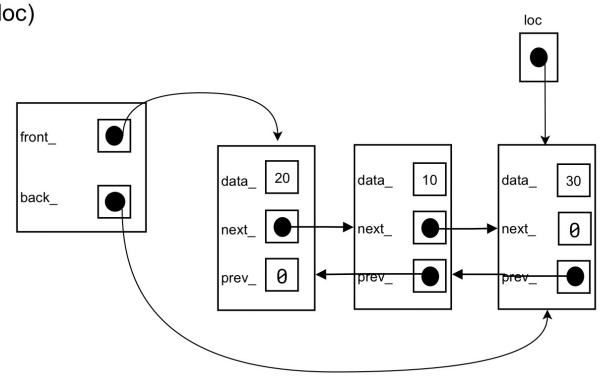
list.erase(loc)



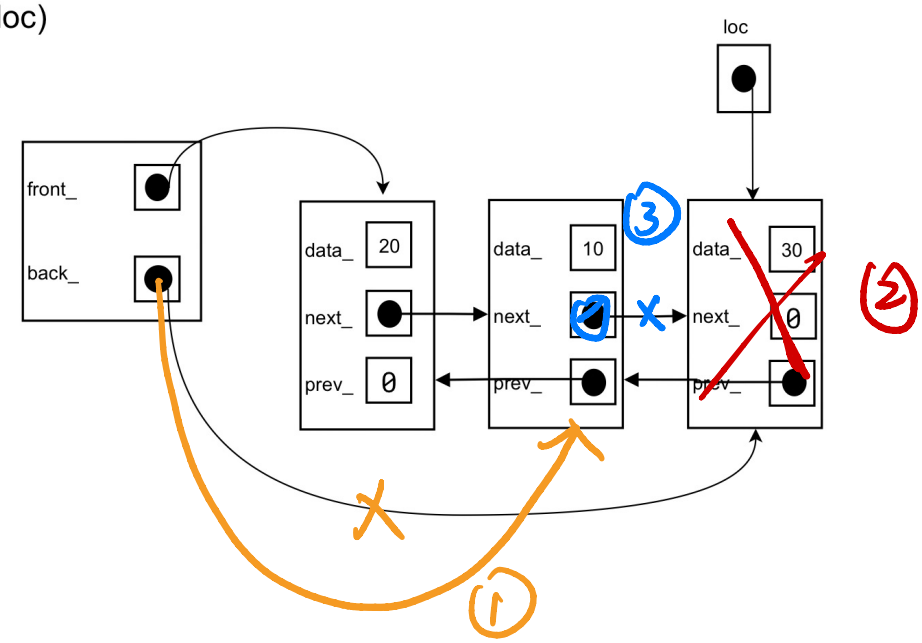

list.erase(loc)
loc
front_
back_
data_
20
next_
prev_
0
data_
10
next_
data_
30
next_
0
prev_


list.erase(loc)
loc
front_
back_
data_
20
next_
prev_
0
data_
10
next_
prev_
data_
30
next_
0
prev_
1
2
3

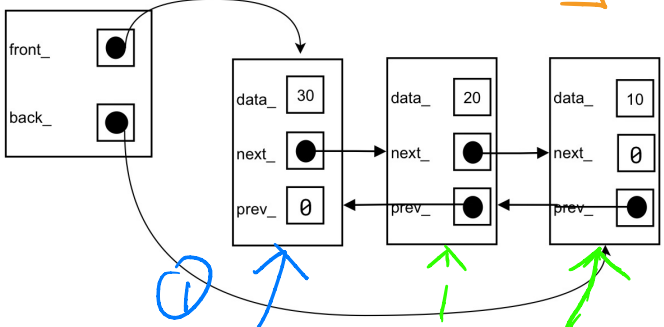
list.erase(loc)



list.erase(loc)



list.sort(begin(), end())



swap. (3)

compare with Anchored Node, swap if smaller

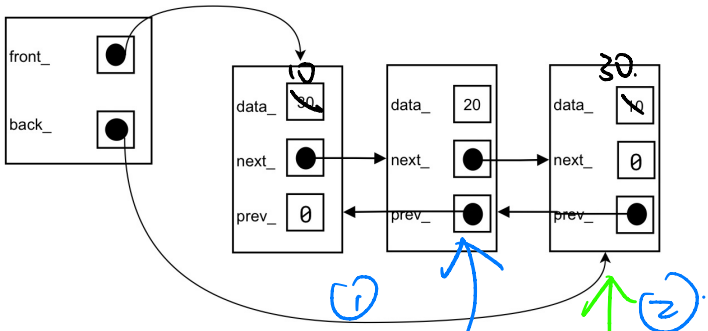
(1)

Anchored Node

smallest node

(2)

list.sort(begin(), end())



(1)

(2)

Anchored Node

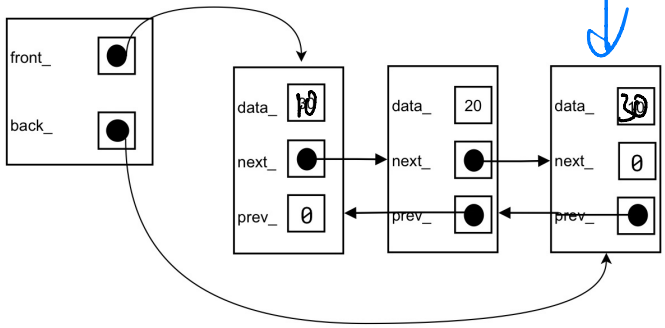
Smallest Node

(3) compare with Anchored Node, no swap

list.sort(begin(), end())

Anchored Node.

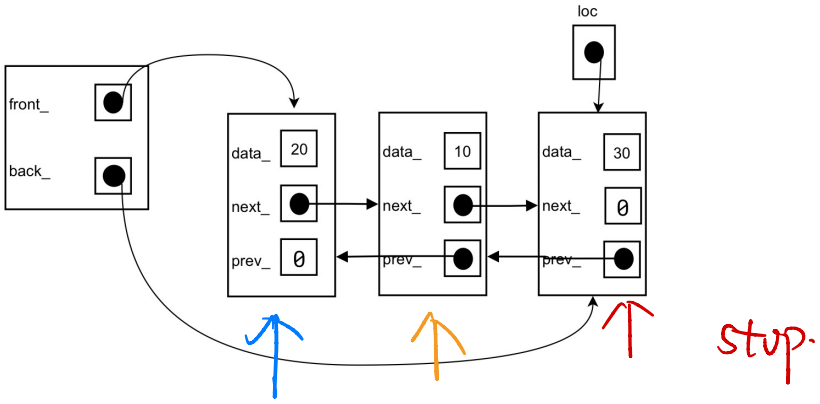
Smallest Node.



null

(3) terminal sorting

list.sort(begin(), loc)

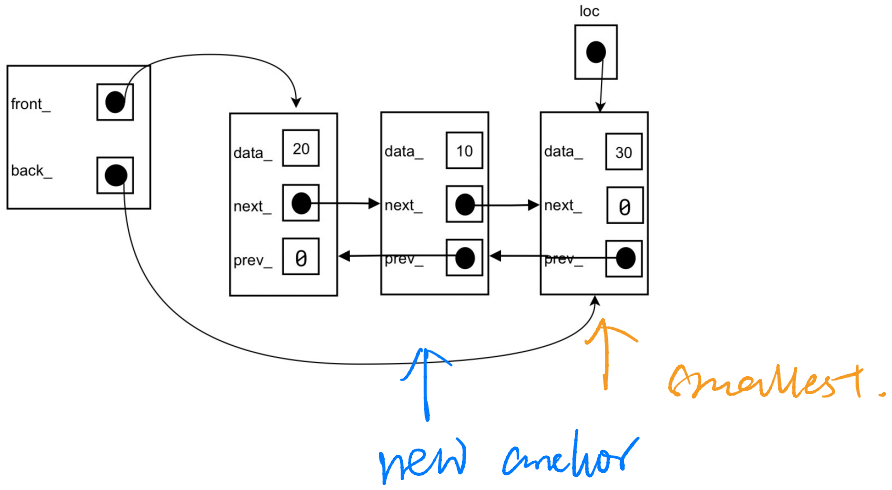


③ compare smallest with every node after anchored.next, update pointer if smaller node is found.

④ when the pointer is at loc. stop.
compare smallest with anchor, if anchor is larger, swap.

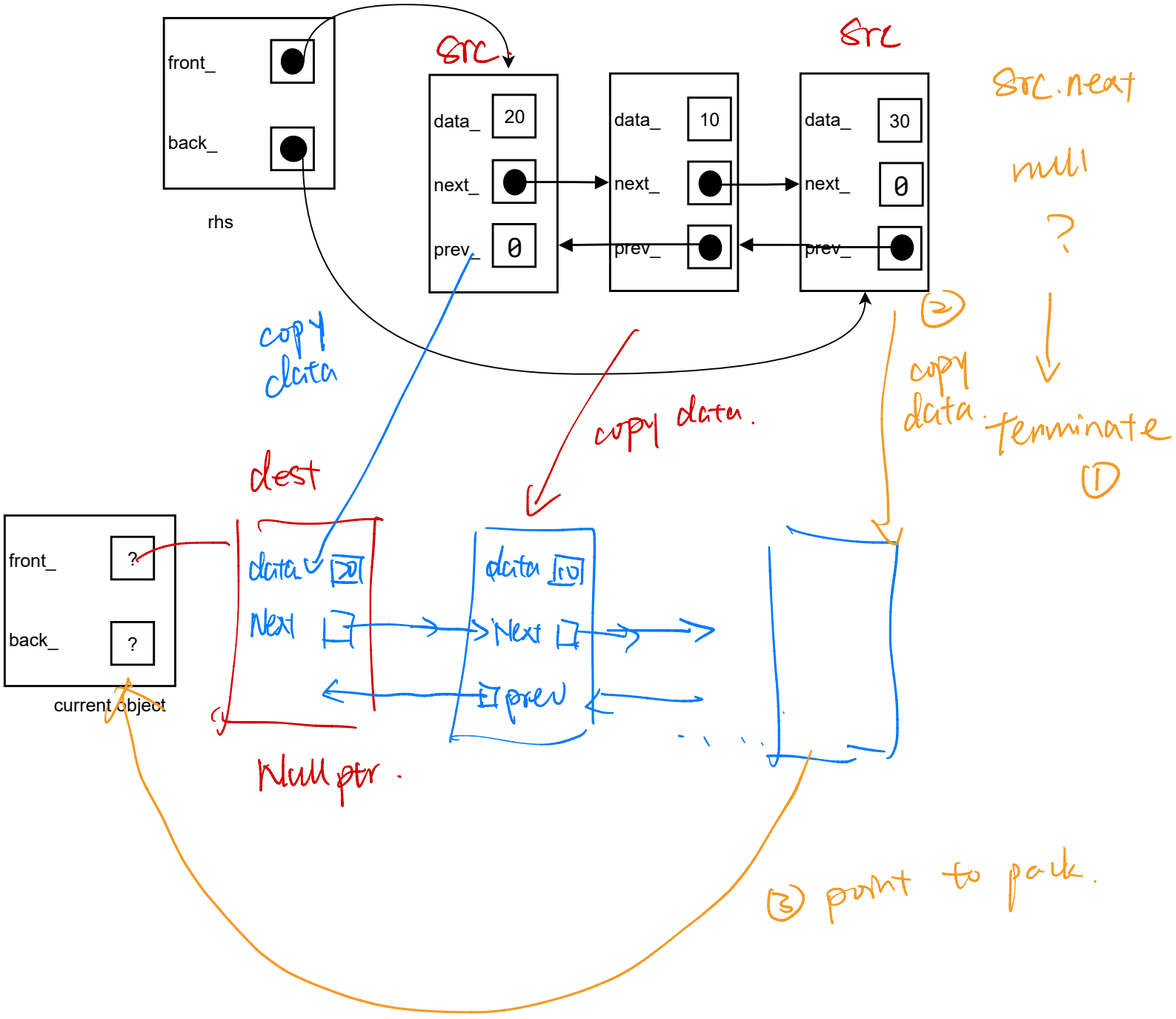
⑤ Advance Anchor, perform the same.

list.sort(begin(), loc)

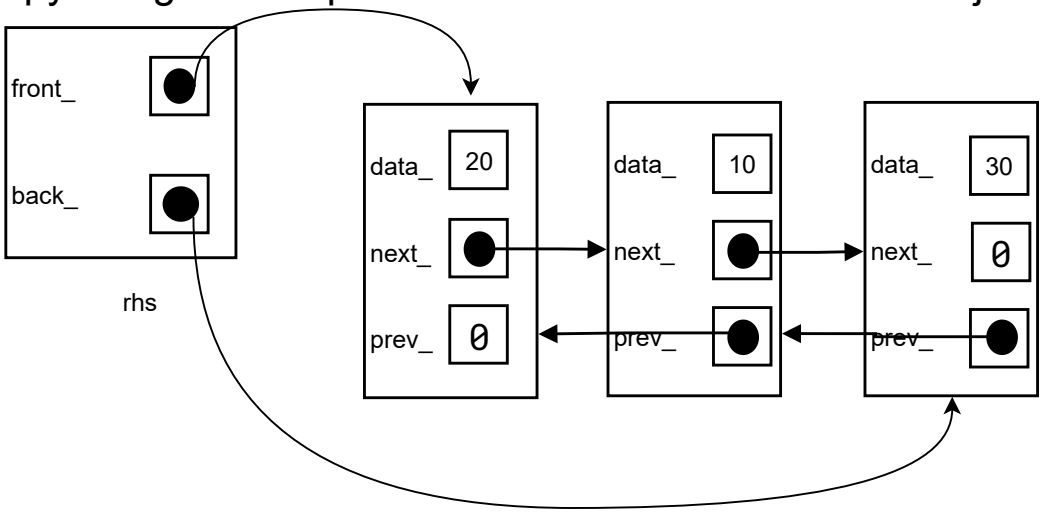


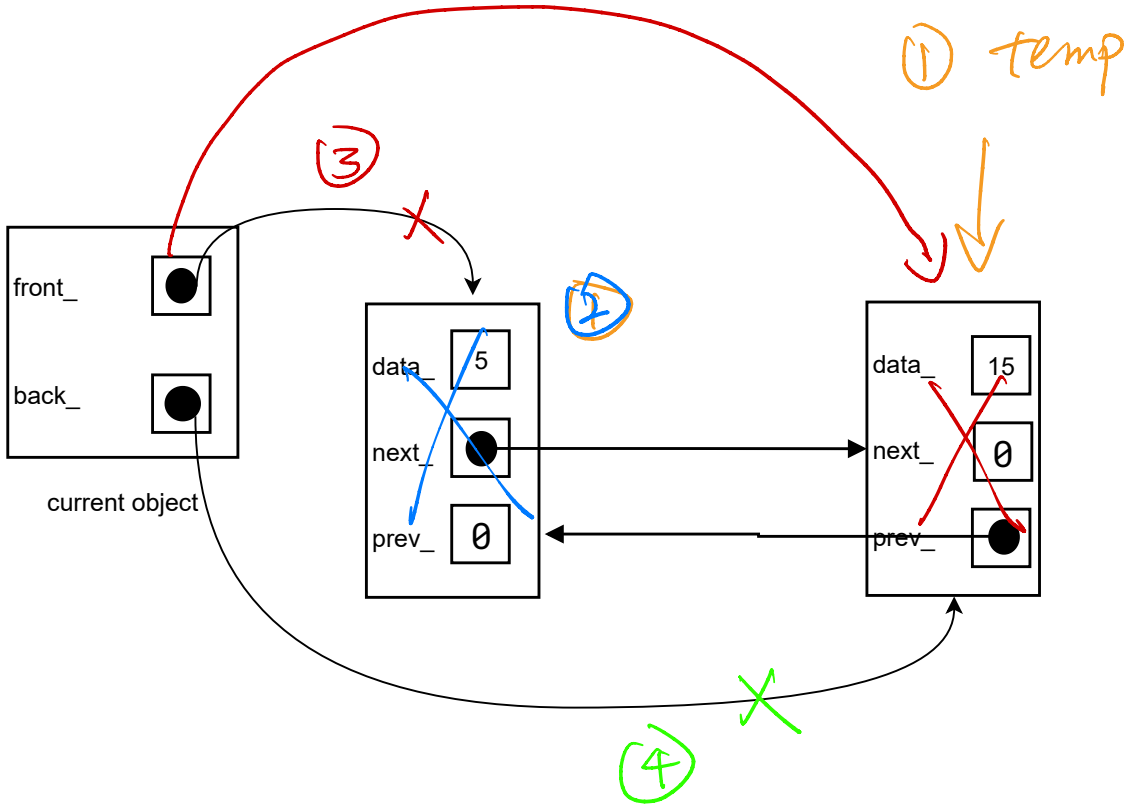
⑥ new smallest is loc. end of sorting.

copy constructor - alter rhs and/or current object as appropriate



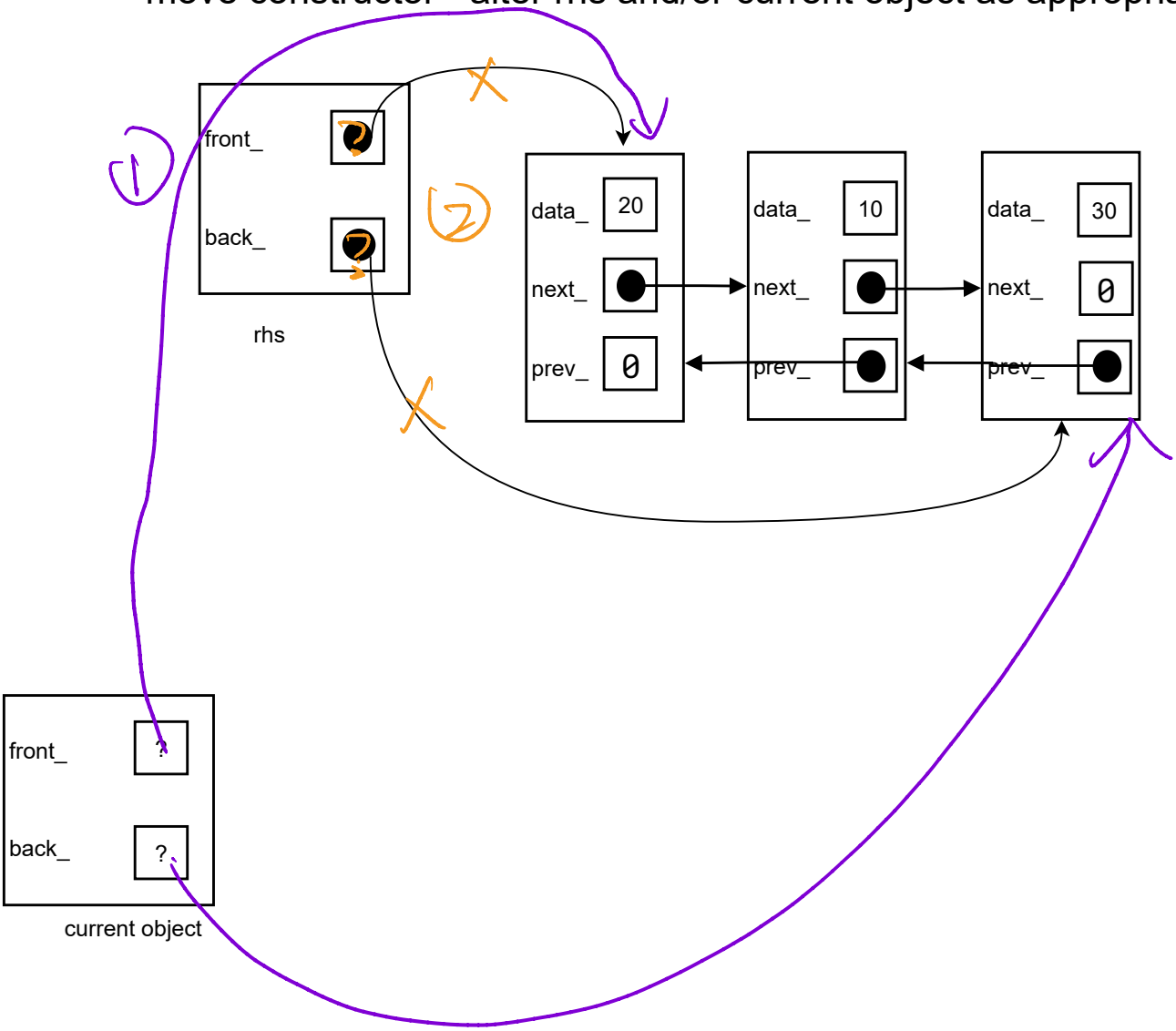
copy assignment operator - alter rhs and/or current object as appropriate



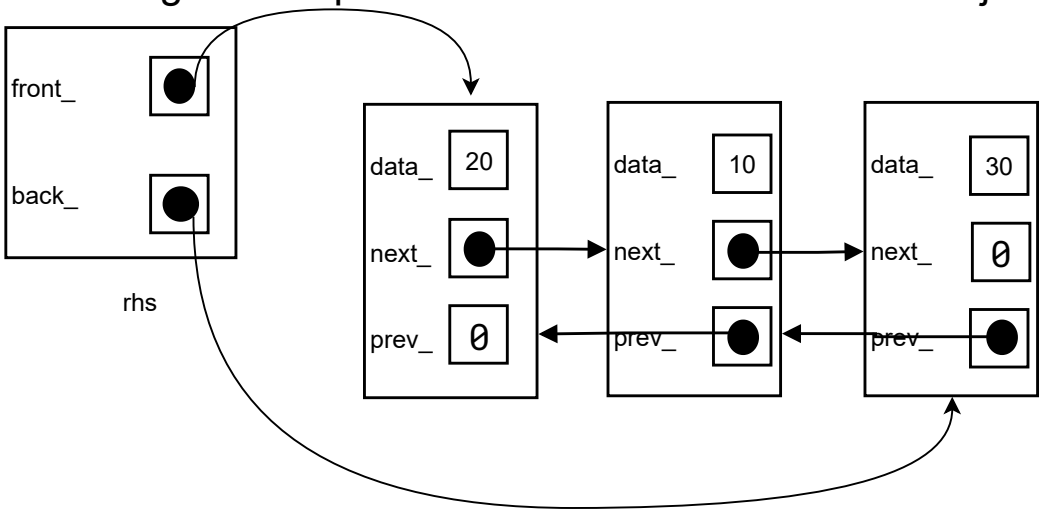


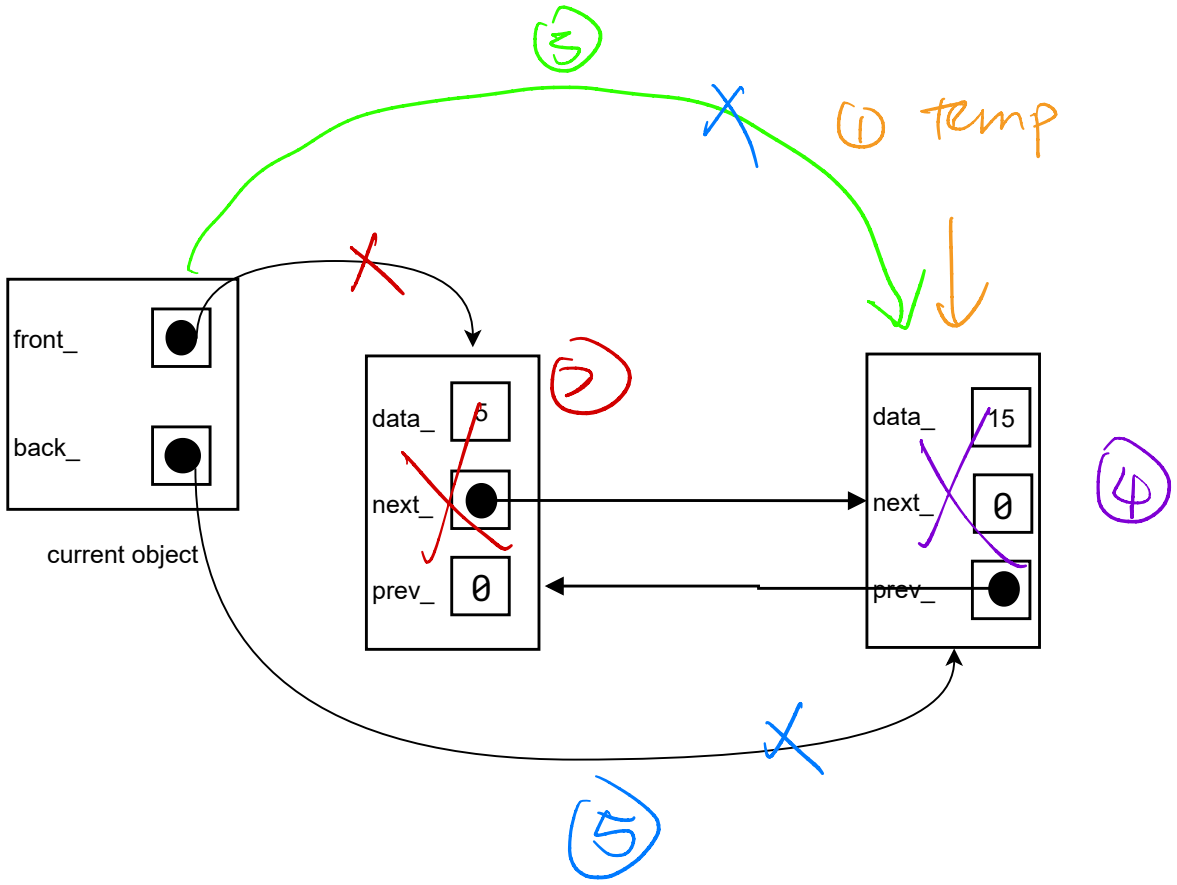
5 copy constructor. part.

move constructor - alter rhs and/or current object as appropriate



move assignment operator - alter rhs and/or current object as appropriate





6 move constructor part