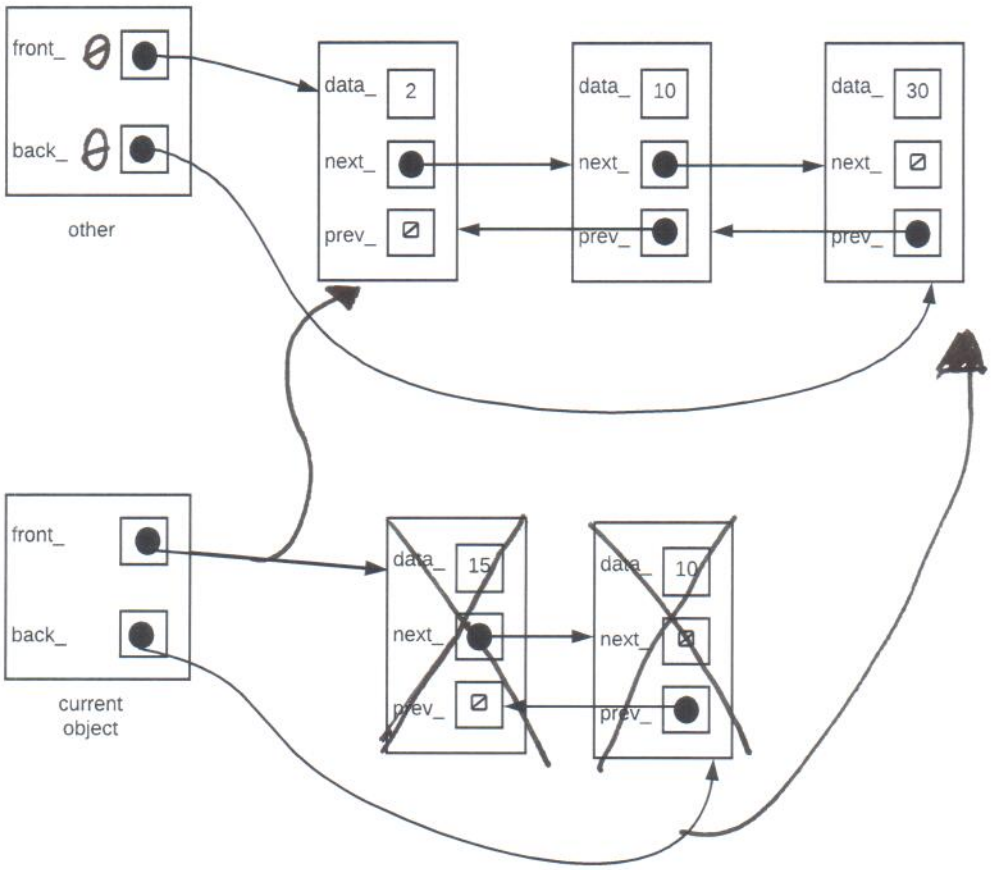
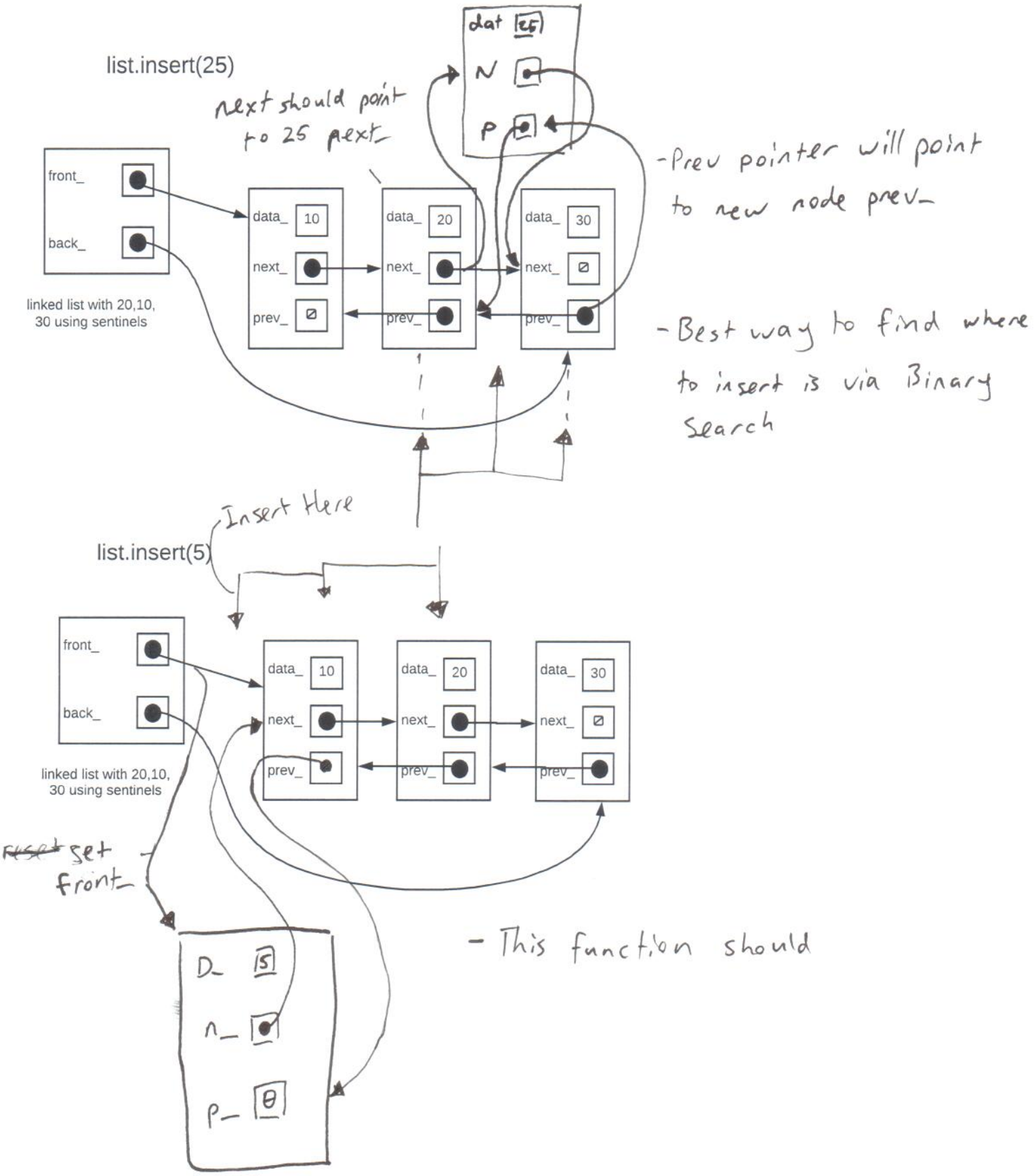
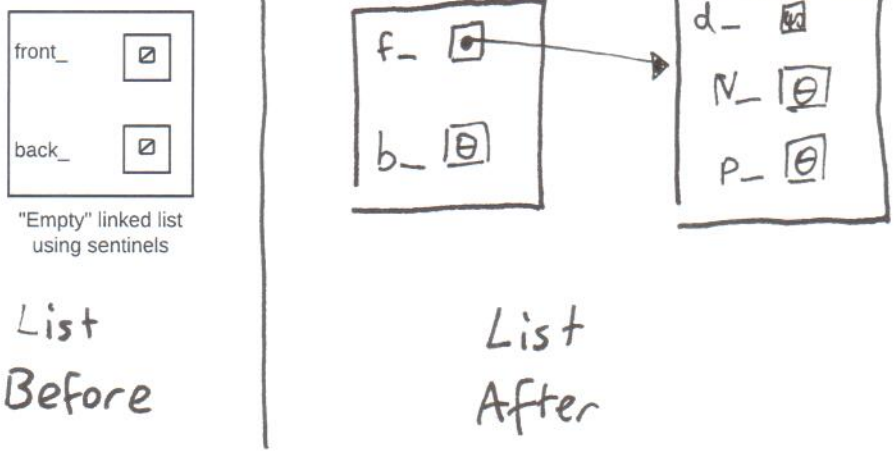
Move Assignment operator




list.insert(25)
next should point to 25 next_
dat
N
P
front_
back_
data_ 10
next_
prev_
data_ 20
next_
prev_
data_ 30
next_
prev_
linked list with 20,10, 30 using sentinels
-Prev pointer will point to new node prev_
-Best way to find where to insert is via Binary Search
Insert Here
list.insert(5)
front_
back_
data_ 10
next_
prev_
data_ 20
next_
prev_
data_ 30
next_
prev_
linked list with 20,10, 30 using sentinels
set front_
D_ 5
N_
P_ 0
- This function should

list.insert(40)

Generate node and set as front

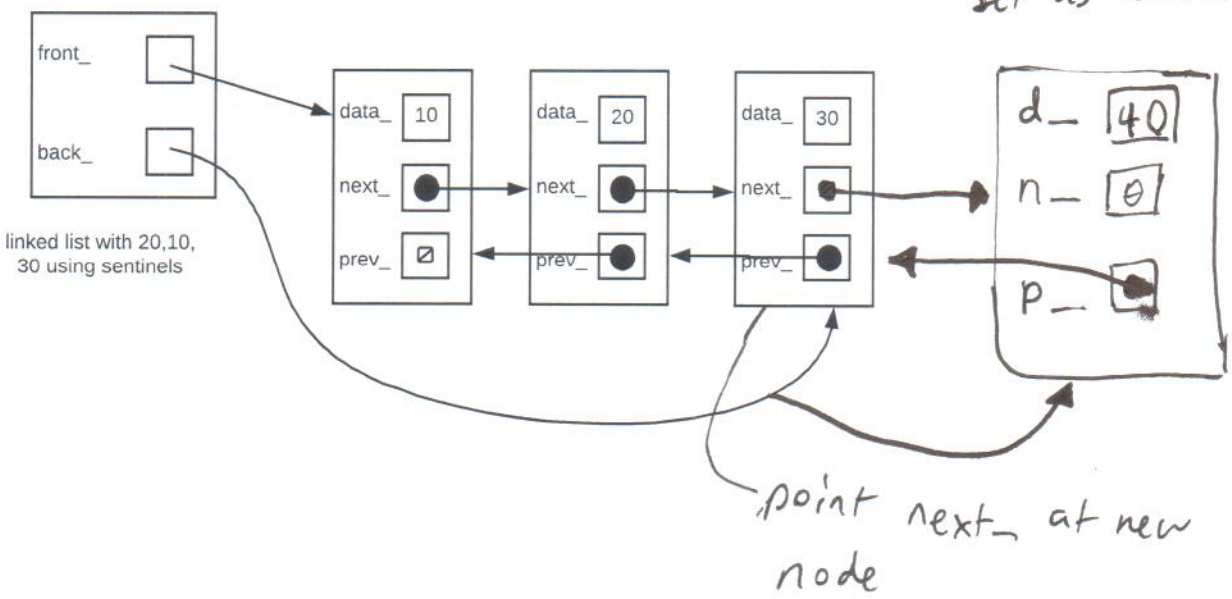


List Before

List After

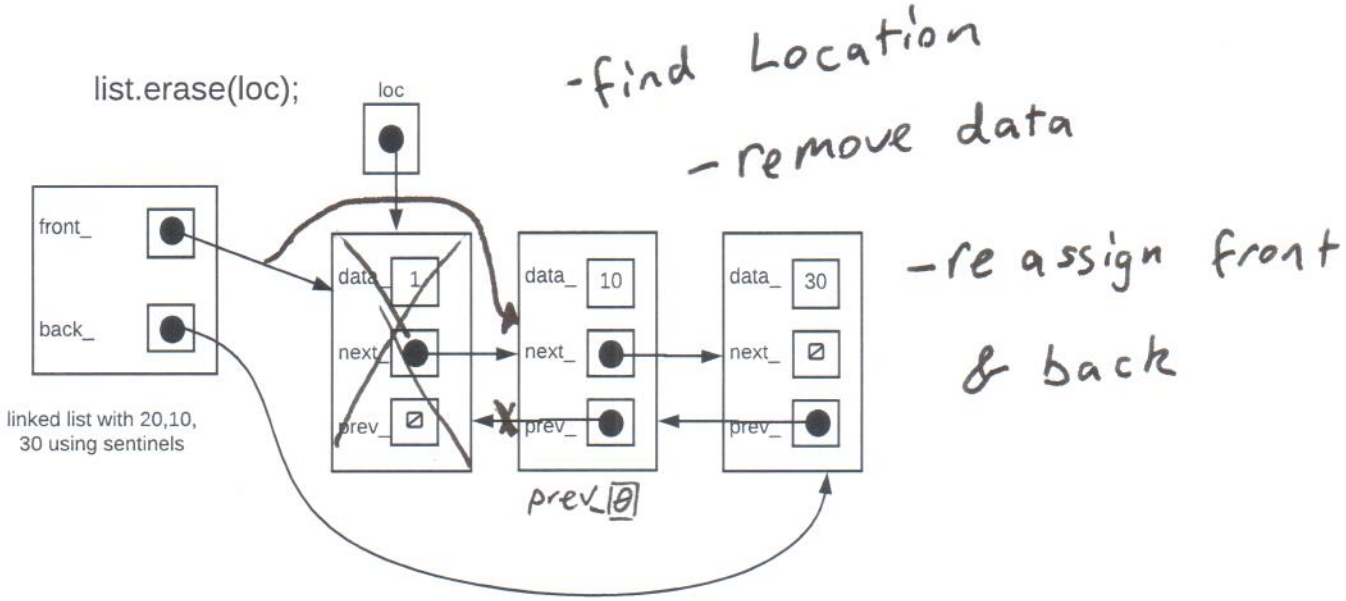
list.insert(40)

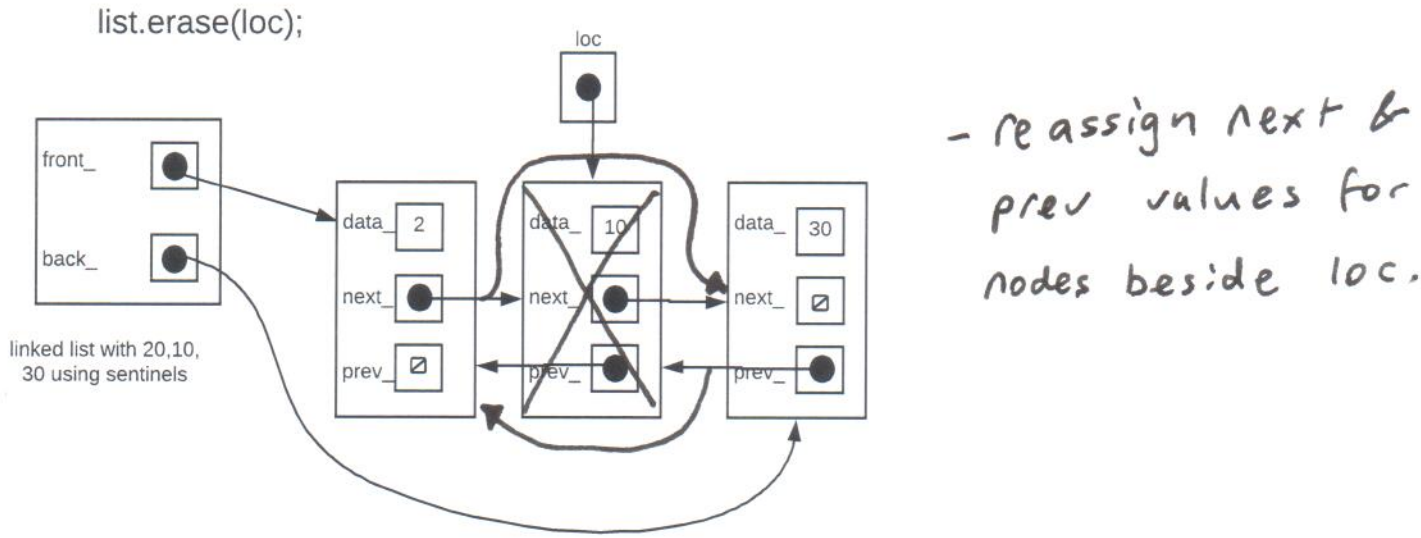
Generate new node set as back_



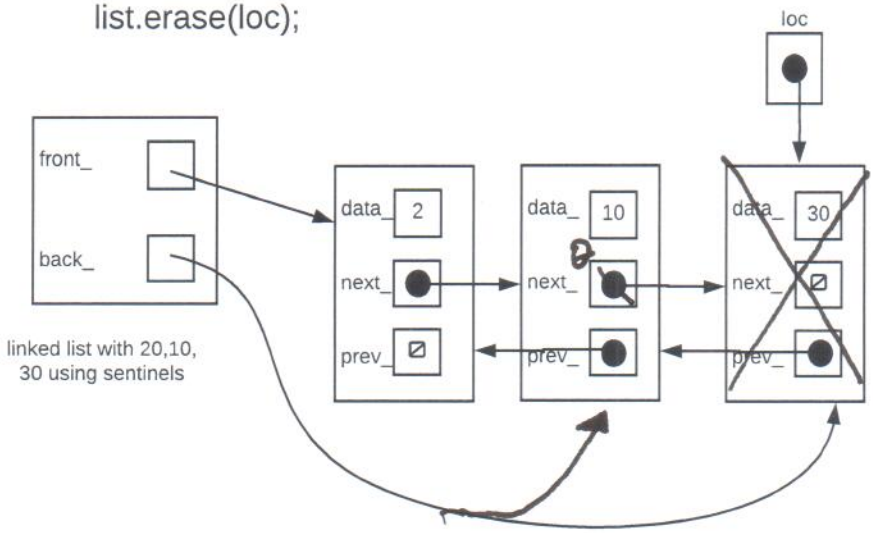
-prev_ for new node will point at previous node.

point next_ at new node


list.erase(loc);
loc
front_
back_
data_ 1
next_
prev_
data_ 10
next_
prev_
data_ 30
next_
prev_
linked list with 20,10,
30 using sentinels
prev_[0]
-find Location
-remove data
-reassign front
& back

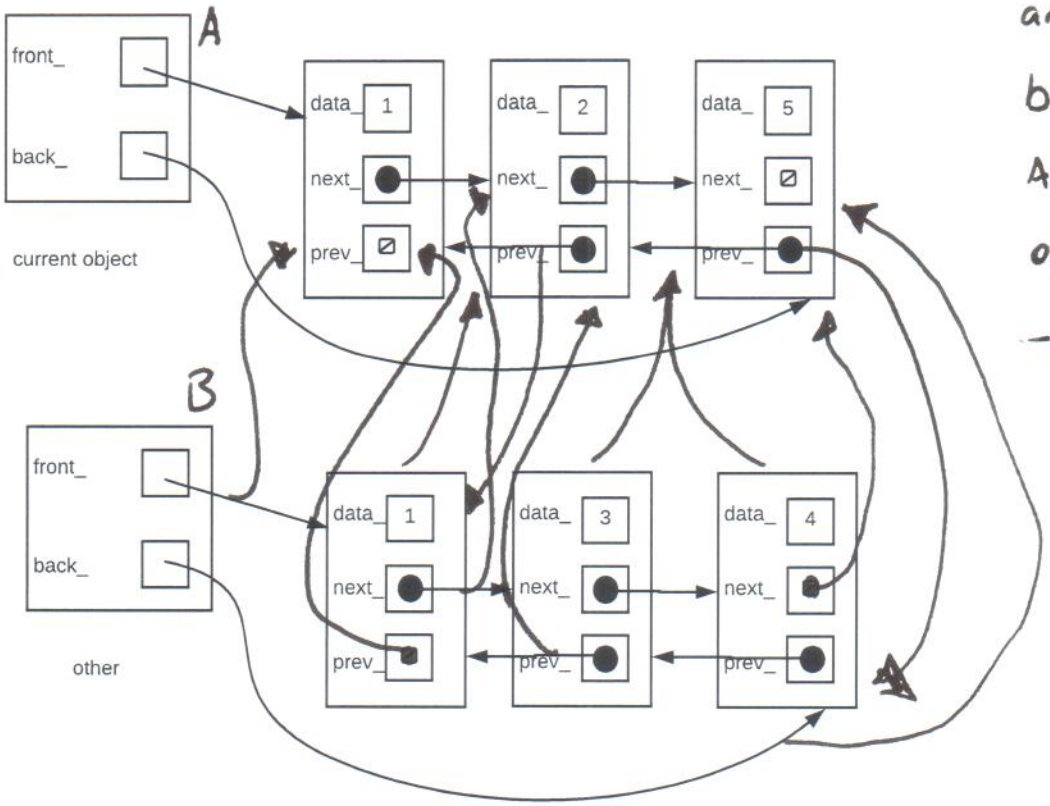

list.erase(loc);
loc
front_
back_
data_ 2
next_
prev_
data_ 10
next_
prev_
data_ 30
next_
prev_
linked list with 20,10,
30 using sentinels
-reassign next &
prev values for
nodes beside loc.

list.erase(loc);

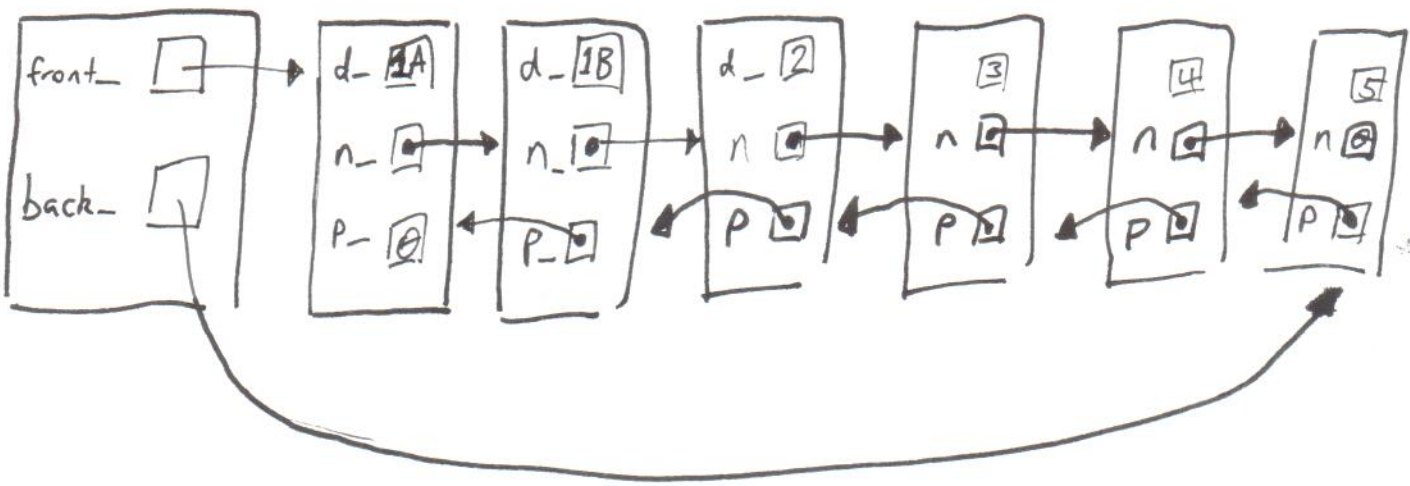


- remove loc after reassigning next_ value.

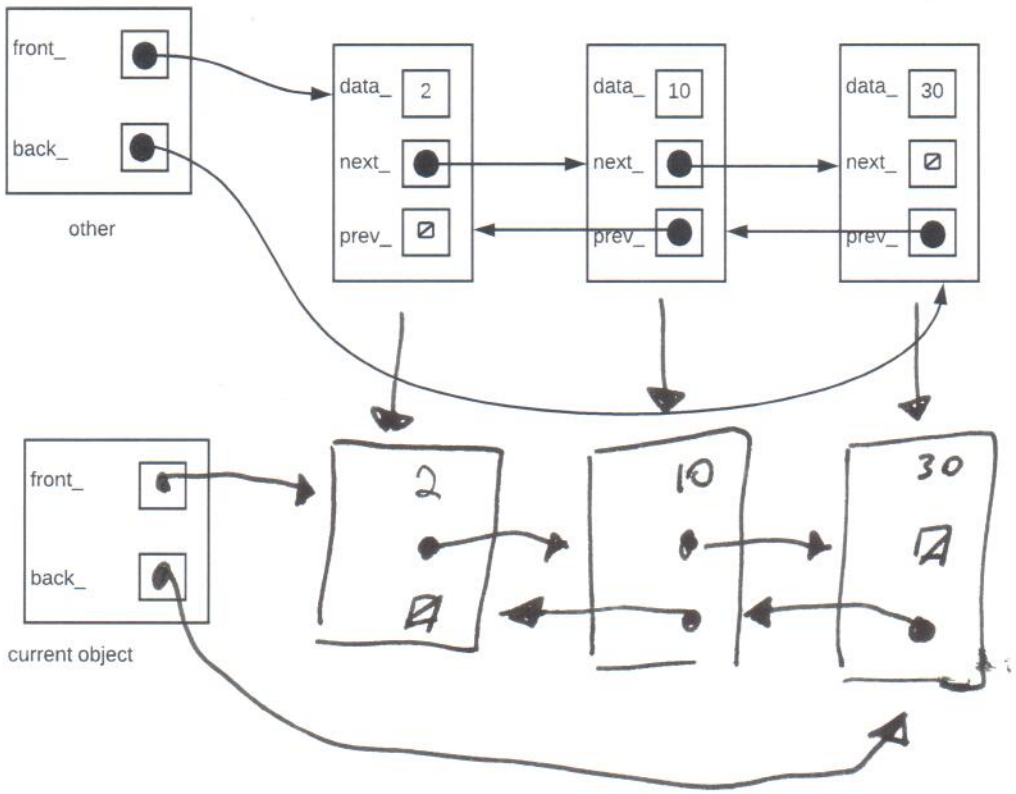
merge(other)



- Assign next_ and prev_ values based on Asrending order of data.
- Only edit the next_ and prev_ values to point to the appropriate data.



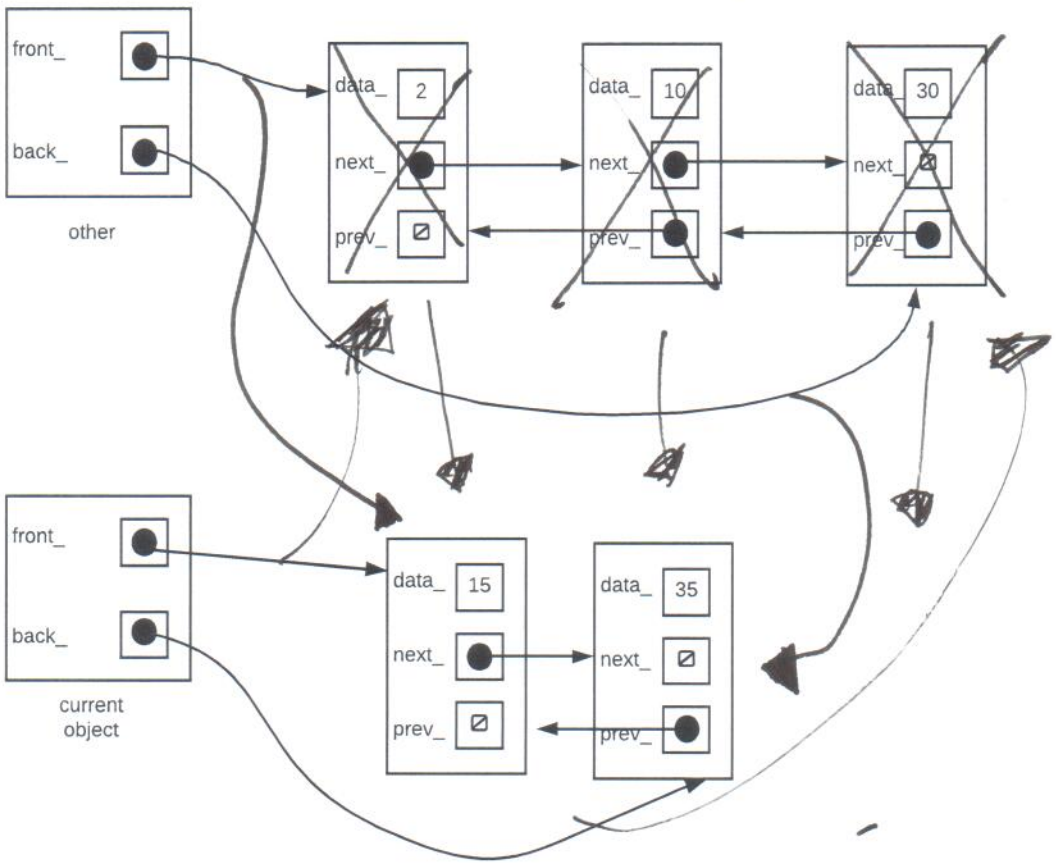
Copy Constructor

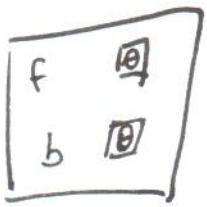


- Standard copy constructor

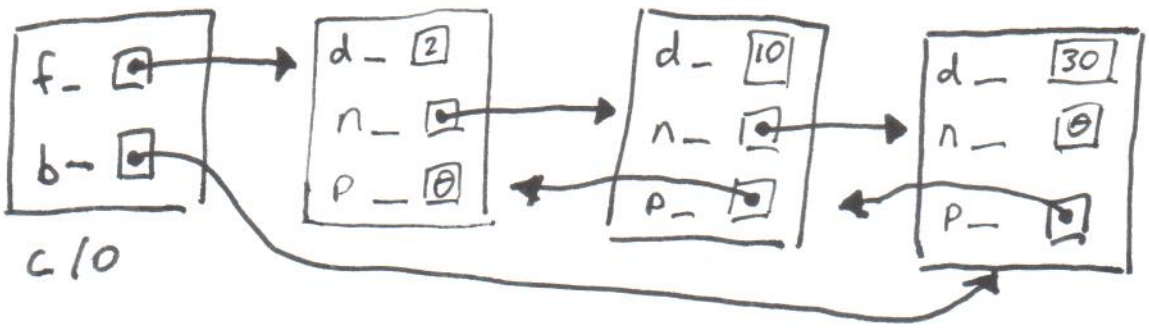
Assignment operator

Incorrect

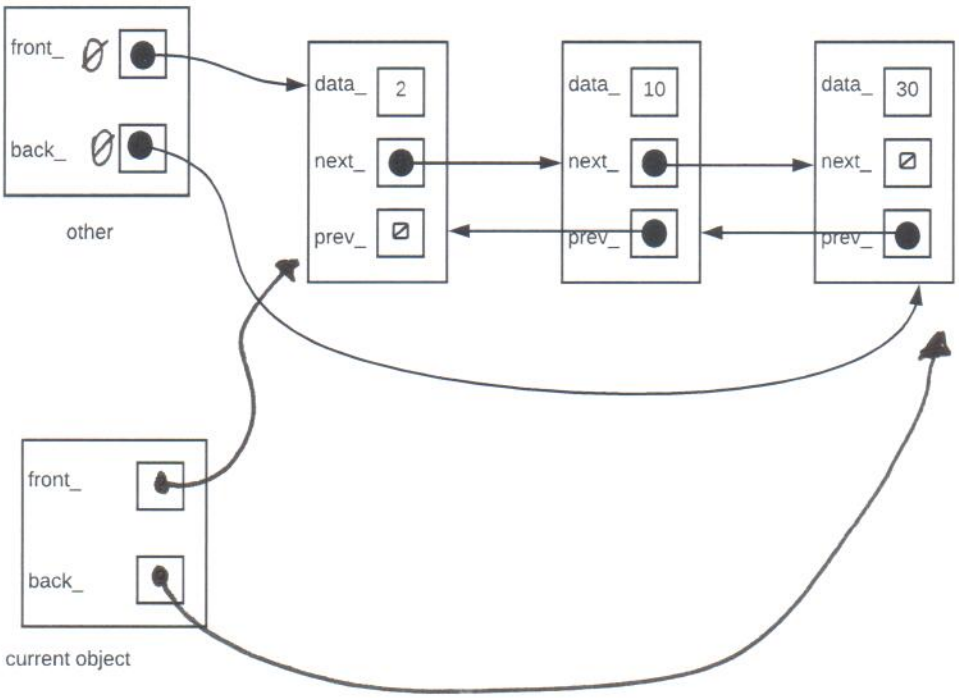


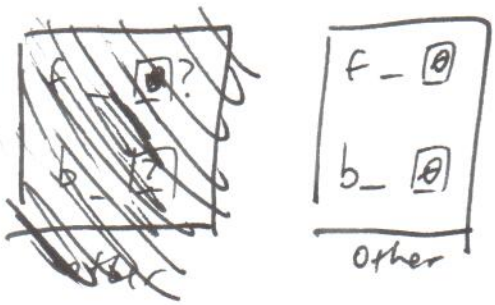


- Other List is deleted
- but first Data from the other list is ~~moved~~
- assigned to the current object
- Current object's nodes are erased before being assigned the other object's nodes.



Move Constructor





- Make temp copy of other in curr. obj.
- Other should register as empty while in use by curr. obj.