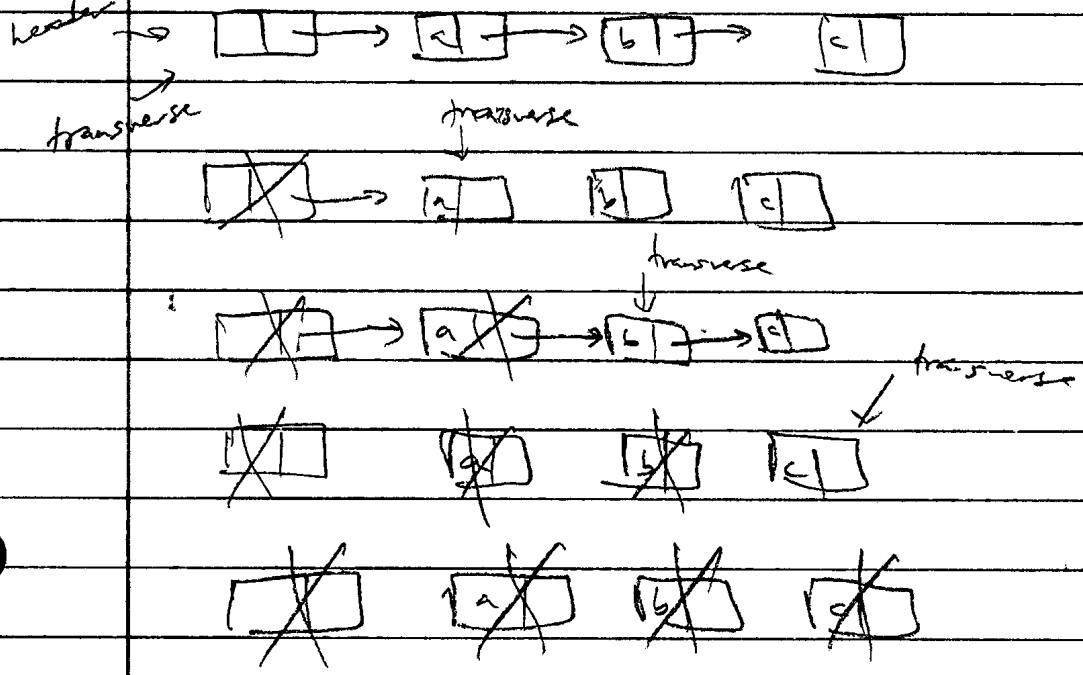
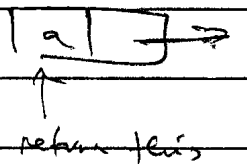
Mohammed Tashid Chowdhury
mtc405@nyu.edu

1. destructor: calls the clear method which deletes each node by looping through the list then sets head to nullptr.

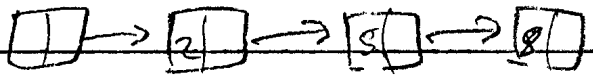


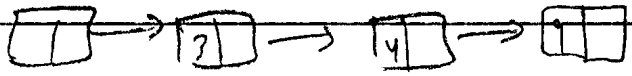
front: returns the data in the first Node

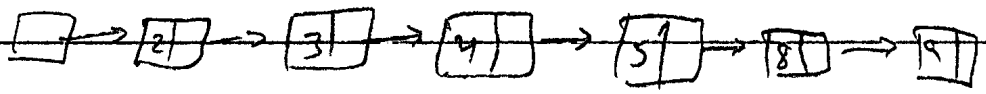


Merge:

while both list not empty
Compare each element of the list and see which smaller. if one is smaller add it to the temporary list and traverse that list to next node.
When one list finishes you copy over remaining elements.
then you copy from temporary list back to original

List 1 

List 2 

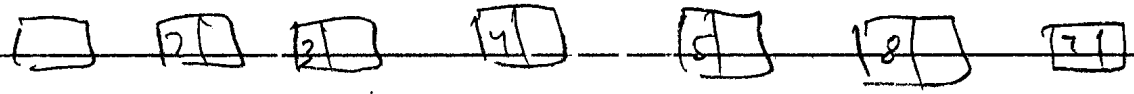
List temp 

erase list 1

copy over temp to list 1.

erase list 2

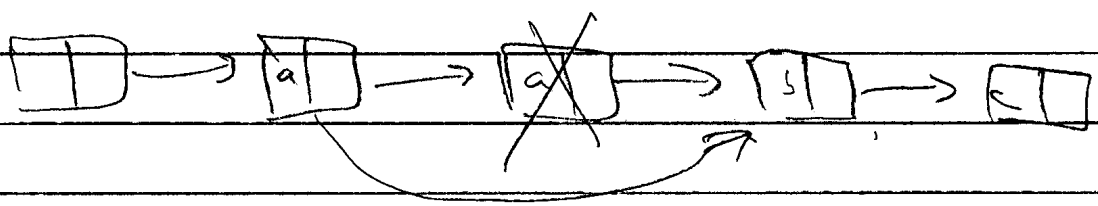
final will look like

list 1 

list 2

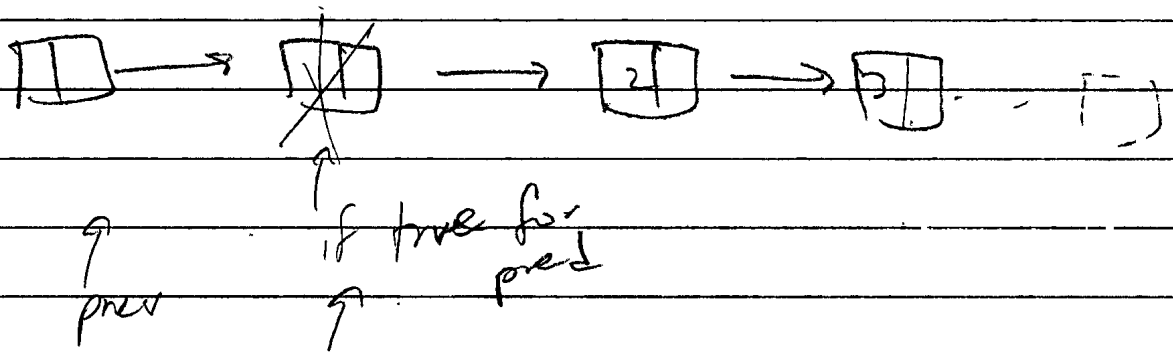
Remove Adjacent Duplicat

- check if trailing isn't nullptr and trailing → next isn't nullptr
  check if trailing → next. = trailing → next → ne
  if it is, call erase after on trailing
  and move trailing forward.

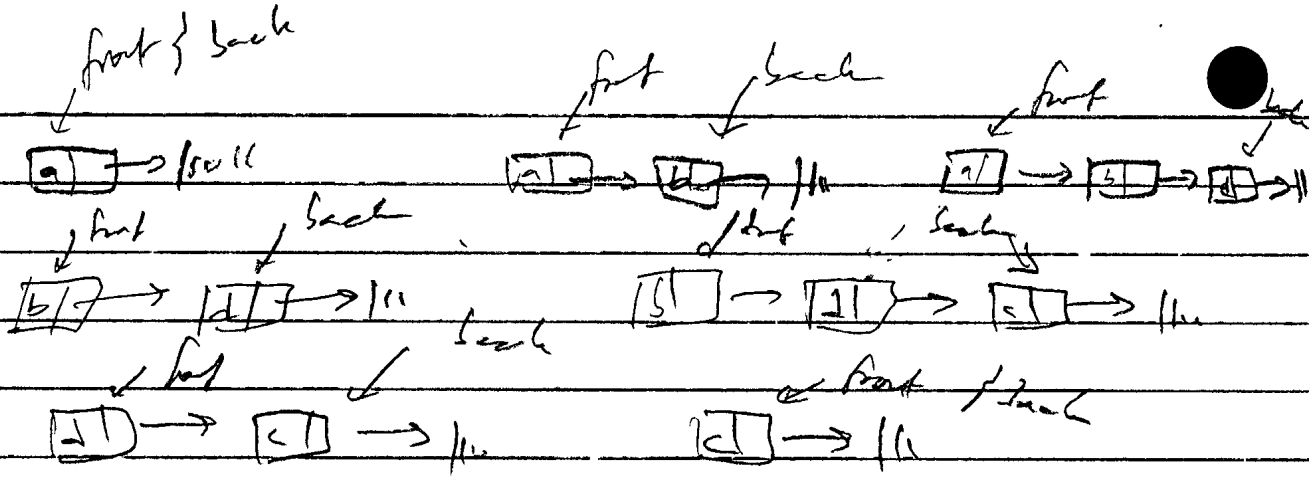


Remove_if.

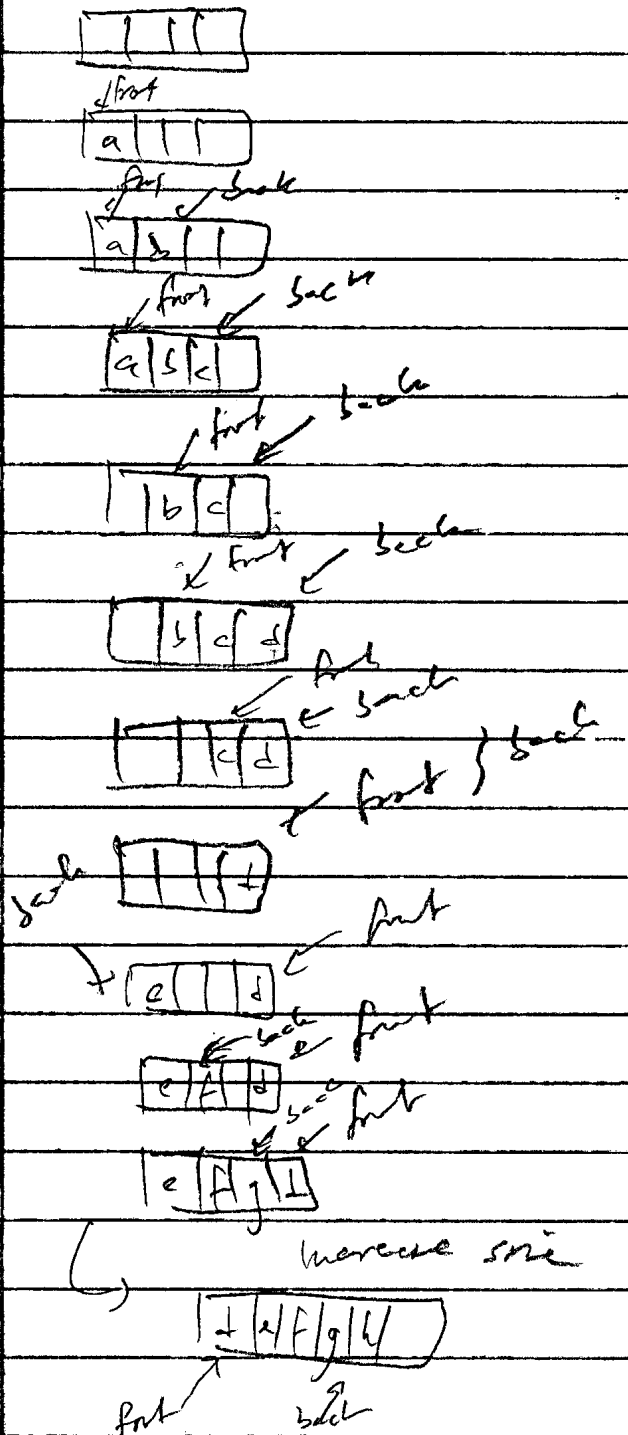
loop through list if the pred of next element is true, then call the erase after on the previous else move both of the forward.



4.


front & back
front
back
front
back
front
back
front
back
front & back

5.


front
front
back
front
back
front
back
front
back
front & back
back
front
back
front
back
front
increase size
front
back