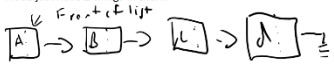
Chapter 17 Note

Remove or delete a node:
Node just have 2 things in them.



Delete node A
Frontoflist.next()
Moves the front of the list to the next node pretty much deleting the node in java

Delete node B
Frontoflist().next = frontoflist().next.next

Remove or delete a node
Current.next = current.next.next

List iterator: call a class that would take us to the next one. 3 things they want to do
-Return things about the current item
-Go to the next one
-Go to the previous one

bigOh time
Insert = constant
Delete = constant
Find = linear
Findprevios = linear

Double linked is bad for memory doubles it.
Double linked list BigOh time is now constant
This is called the memory time trade off.
Insert will slow down a bit because you have twice the arrows to repoint.

Circularly linked list
Don't have front or back
Could do doubly linked list or single linked list with circularly linked list

Sorted linked list
Methods that need to change add and delete
Can just add to front have to change, BigOh time would be N

When would you use a linked list
Anywhere you would use an array
You can import linked list
Making an image view/music player/ file viewer
In stack and queues
Modeling a line at the store.

Use linked list over array if you want to add things to the front

17.6 hw help