Resources used to help with assignment: <http://algorithms.tutorialhorizon.com/reverse-a-linked-list/>

Documentation

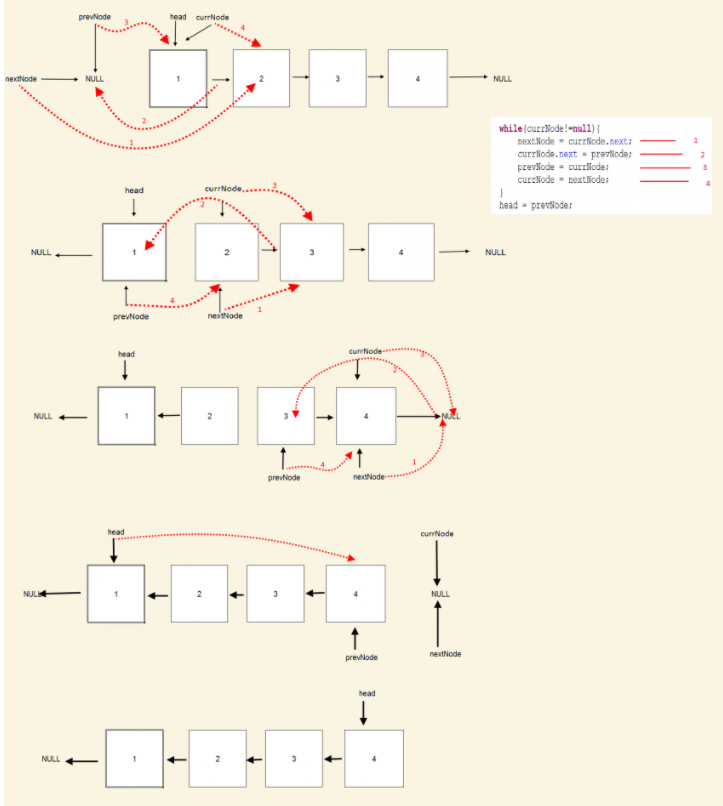
1. Discuss your design and approach for solving the problem

I used the SList class in the previous program assignment, program 5, as a base to build my list class. I made a list of all the functions I would need to implement for my list from the SList class and made a note of any other methods I would need like the goToPrevNode, reverseRecursive, and reverseIterative.

For the insert method I had to make sure to specify what happens in the case if the list is empty, if the insert is to happen at the end, or if the insert happens in the middle. The same consideration was given to the remove method.

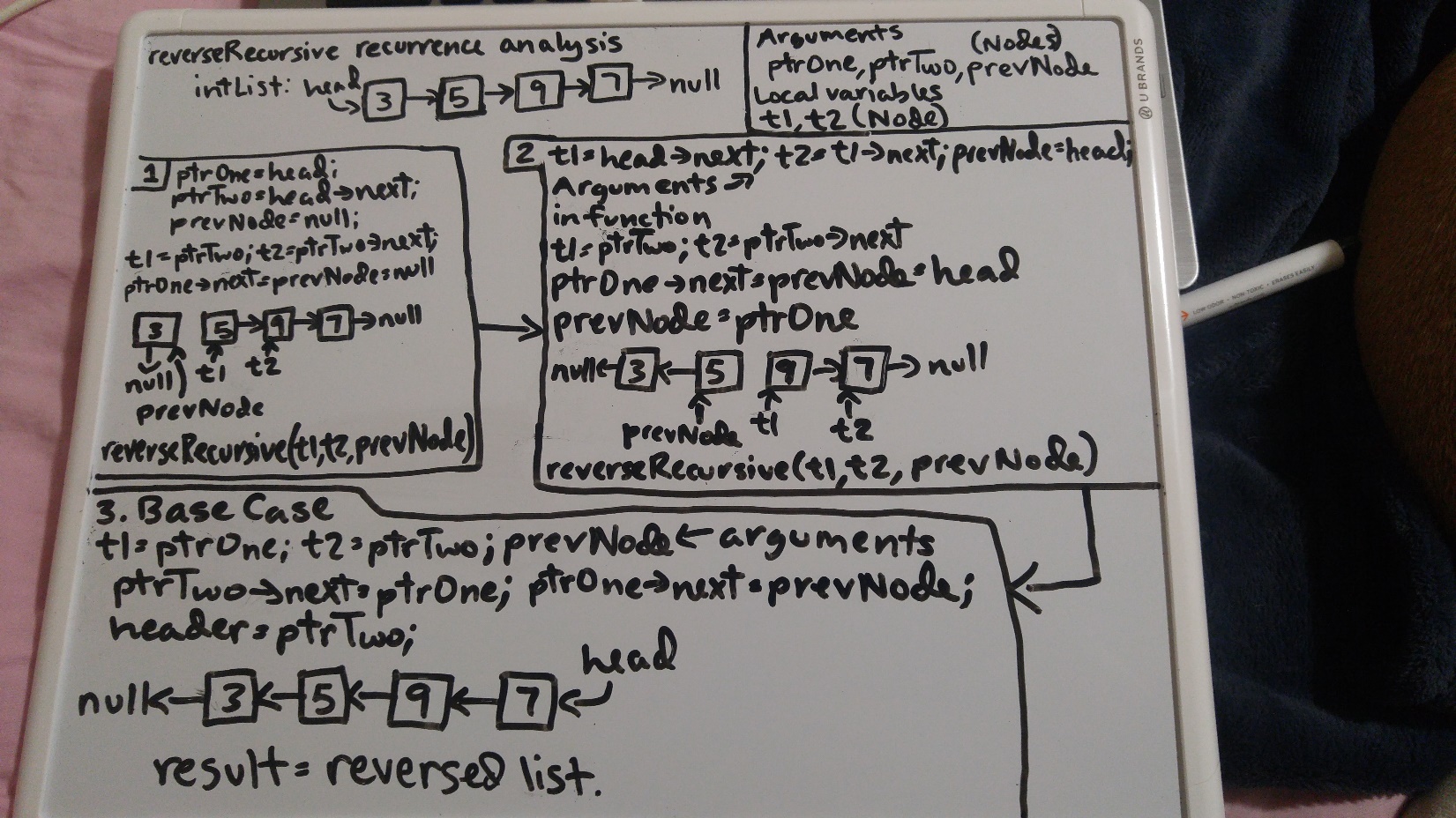
I used the link provided to code the reverse recursive and iterative.

1. You are encouraged to use pictures to illustrate what you're doing



The above picture shows the process of reverse iterative and was gotten from the site linked at the top of this word doc.

Below is my recurrence code analysis of the reverseRecursive method.



1. Discuss the differences and similarities between the two approaches, i.e., iterative vs. recursive

Both the solutions to reverse a list, iterative and recursive, are fairly similar. Each iteration of the while loop in the iterative solution reverses the next link of current node to the previous node so in the first iteration the next link to current node, which is the first node, will be set to null. For each call of the recursive function 3 arguments are passed in, ptrOne, ptrTwo, and prevNode, and the next link of ptrOne is set to prevNode which in the first iteration sets the next link of the first node to null. The local variables t1 is set to ptrTwo and t2 is set to ptrTwo->next and prevNode is set to ptrOne and then t1, t2, and prevNode are used to make each successive recursive call.

The differences are that the base case in the recursive solution changes the next link of the last two nodes in the list completely reversing it whereas the last iteration of the while loop in the iterative solution only has to reset the next link of the last node which completely reverses the list. Another difference is that the recursive solution requires extra memory because it initializes 2 local variables, t1 and t2, that are used to make each successive recursive call.