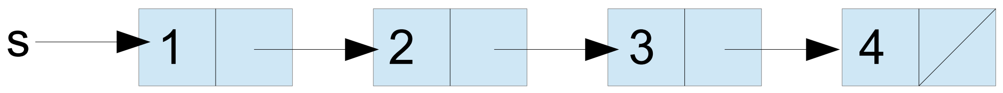
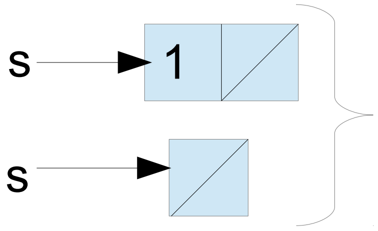
Here's the intuition behind solving the problem:

Here's the initial linked list.
We need to return 4, 3, 2, 1 by modifying the list that is given.

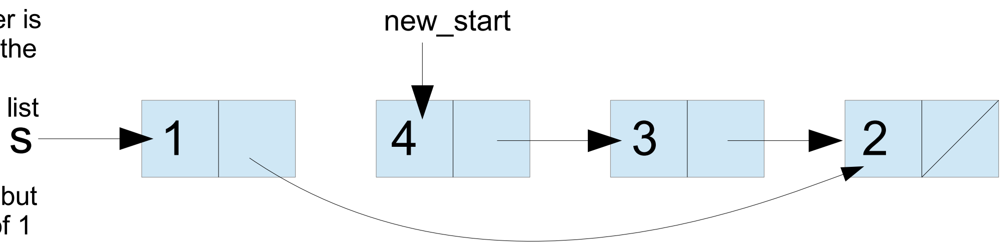


First do the base cases:
1. If the initial list is empty we just return Link.emtpy
2. If we pass in a list with one element (rest is empty) we return just the list with no modifications
Note that in both of these cases you return whatever you passed in.



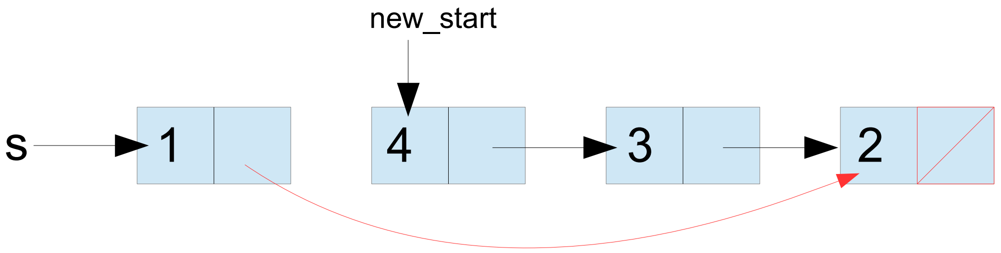
just return s

Now do the leap of faith:
Say we get to this point.
Note that new_start is whatever is returned by calling reverse on the rest of the list. It will return the new beginning of the reversed list of rest.

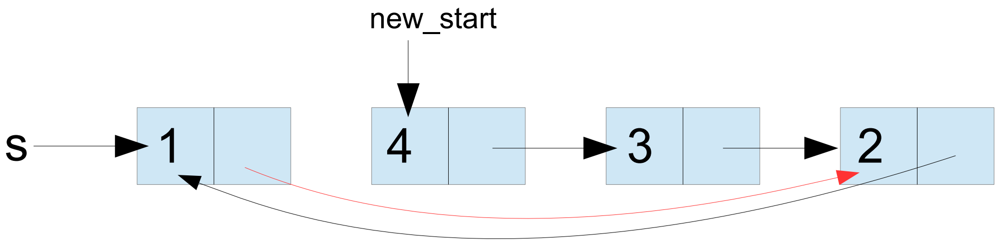


The rest of the list is reversed but s still points to 1 and the rest of 1 still points to 2.
It is vitally important to understand why s and s.rest point where they do (the next step relies completely on this fact)

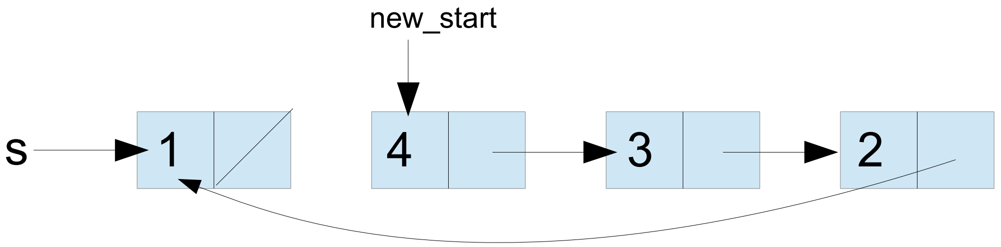
The things in red are what we have to fix to return a completely reversed list.



First fix the rest of the last element (s.rest.rest)
To see why this is s.rest.rest follow the red arrow going out of the first box.



Now fix the rest of 1. We want 1 to be at the end of the new reversed list, so its rest should just be Link.empty



Return new_start and you're done :D