Data Structures

Directions

- DO NOT add new import statements.
- DO NOT add any new class attributes
- DO NOT change any of the method's signatures.
- DO NOT modify the given constructor.

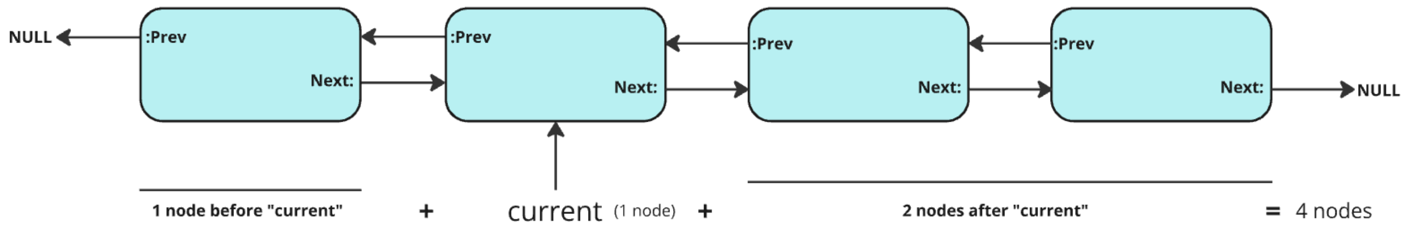
To complete this Lab, implement the **numberSites()** and **addRecent()** methods for the doubly linked list of strings. Each node in this doubly linked list is represented as a WebPage, with a String name, a next field and a prev field. The next and prev field both point to another WebPage node.

The method signatures are already given, do not modify these or add any new ones.

There is one class attribute provided for you in DoublyLinkedWebPage.java:

- "**current**" which is a reference to a single node in the doubly linked list
  - This is **not** a reference to specifically the front or end of the list. Since the list is doubly linked, the driver and tests move this pointer back and forth along the list. This differs from past examples you've done in class.
  - current is the currently viewed page in the driver.

numberSites()



This method is a counter method, which returns the total number of nodes that are currently in the list. Use a WebPage pointer variable to traverse the list, and use an integer counter to count the number of nodes.

REMEMBER, the list is **Doubly** linked and the "current" attribute may point to any node in the list. SO, to count the total number of nodes, you need to start at "current" and count in **BOTH** directions (both next and prev, aka forwards and backwards).

- OR you can find to the front of the list and start counting from there.

Note: Make sure to use a pointer variable to do this traversal, do not change the "current" attribute. Meaning that "current" does not move, so you need an auxiliary pointer variable (just like we did in class).

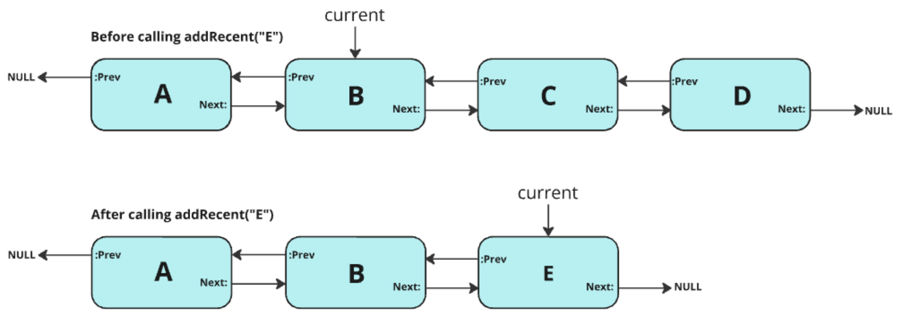
There are **NO** getter/setter methods provided in the WebPage class. To traverse, directly access the "next" and "prev" attribute with "ptr.next" and "ptr.prev" (where "ptr" is a variable that holds a WebPage node).

Once you've traversed the whole list, return the number of WebPages.

addRecent(String toAdd)

This method adds a new WebPage node with the given string immediately after "current" (aka current.next). It does not keep the nodes that were originally after current, so it effectively deletes them.

- This is done to mimic the behavior of the forward/back buttons on a web browser. When visiting multiple sites in succession, they all get added to a list that you can traverse with the forward/back buttons. But if you move back, then click a new link, you no longer can move forward as you have added a new recent site.



To complete this method:

-When the list is empty (current == null), then set current equal to a new WebPage node, and set its website to the given String.

– If the list is not empty, create a new WebPage node, and set its website. Then, set its "prev" attribute equal to current. Then, set current.next to that new node. Finally, current equal to current.next.

Hint: You do NOT need to use any loops to complete this method.