

Assignment 3: merge sort on doubly-linked list

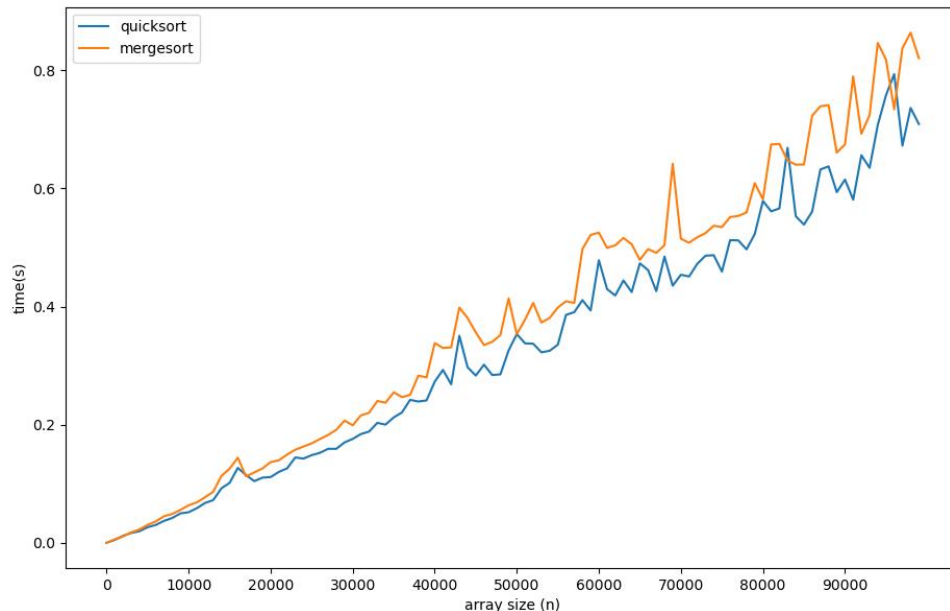
Due: Wednesday, March 9th at 11:59 PM.

Using the class `_DoublyLinkedListBase` from page 274 of lecture 8 notes as the linked list implementation, implement mergesort on a doubly linked list.

Your mergesort function should accept an instance of type `_DoublyLinkedListBase`, which you will construct in main by repeatedly inserting random numbers into a `_DoublyLinkedListBase` instance.

Once you have verified that your sorting algorithm is correct, test the algorithm on a range of list sizes from 10 to 1000 in increments of 10 ($n=10, n=20, n=30, n=40, \dots, n=1000$).

Finally, implement merge sort in python that handles a python List (array type) as input, instead of doubly linked list (can use my implementation on moodle). Compare the sorting times for mergesort on the `_DoublyLinkedListBase` vs the List implementation. Use python's matplotlib plotting function to generate a plot similar to what you see below, comparing mergesort on a `_DoublyLinkedListBase` to mergesort on a List:



*note – the above plot compares quicksort and mergesort, from input sizes of 1 to 100,000. You will show something similar, but compare linked `_DoublyLinkedListBase` mergesort to List mergesort on input sizes of 10 to 1000.

Include your plot as a pdf file along with your source code. Comment on the efficiency of linked list vs array implementation. Which is faster? Why?