# MT 2MD3 Assignment 2 – Michael Giancola

1. In order to swap two nodes x and y in a **singly linked list**, we need to ensure we have references to previous nodes in the list.

To swap x and y in a singly linked list:

- Find the previous node of x (prevX) and find the previous node of y (prevY)
- Check to see if either x or y is the head of the list. If x is the head of the list then set the head of the list to be y and if y is the head of the list then set the head of the list to be x
- Make prevX.next = y and prevY.next = x
- Now we must swap the next pointers of x and y:

```
 temp = x.next;
x.next = y.next;
 y.next = temp;
```

In order to swap two nodes x and y in a **doubly linked list**, we must ensure we have references to the header, and trailer of the list and to x and y.

To swap x and y in a doubly linked list:

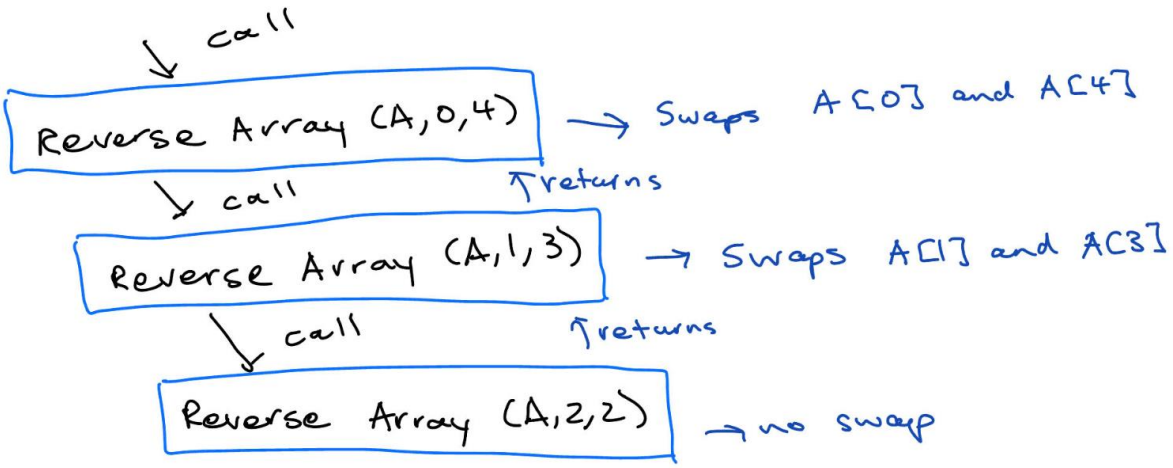
- Similar to the singly linked list, check to see if either x or y is the header of the list. If x is the header of the list, set the header to be y. If y is the header of the list, set the header to be x
- Then, check if x or y is the trailer of the list. If x is the trailer of the list, set the trailer to be y. If y is the trailer of the list, set the trailer to be x
- Now, swap the next and previous pointers of x and y:

```
 temp = x.next
x.next = y.next
 y.next = temp
 temp = x.prev
x.prev = y.prev
 y.prev = temp
```

- If x.next is not None, x.next.prev = x
- If x.prev is not None, x.prev.next = x
- If y.next is not None, y.next.prev = y
- If y.prev is not None, y.prev.next = y

Doubly linked lists are able to swap more efficiently as they have access to previous nodes and work in both directions however, both of the algorithms have the same time complexity (O(1)) and they both perform constant time operations regardless of the size of the given list.

2.




- A[0] and A[4] are swapped on the first call
- A[1] and A[3] are swapped on the second call
- Nothing is swapped on the third call because i is equal to j so the loop isn't entered and the newly ordered array is returned