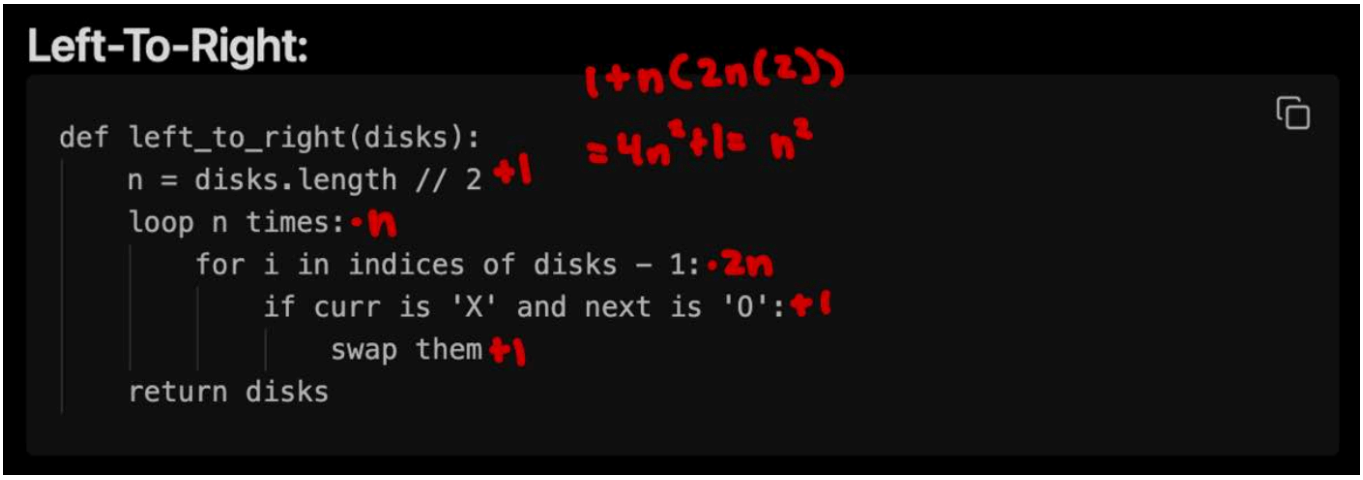
Project 1

Esteban Escartin - eescartin@csu.fullerton.edu

Here is the pseudo code and step counting for the two algorithms:

Left-To-Right:

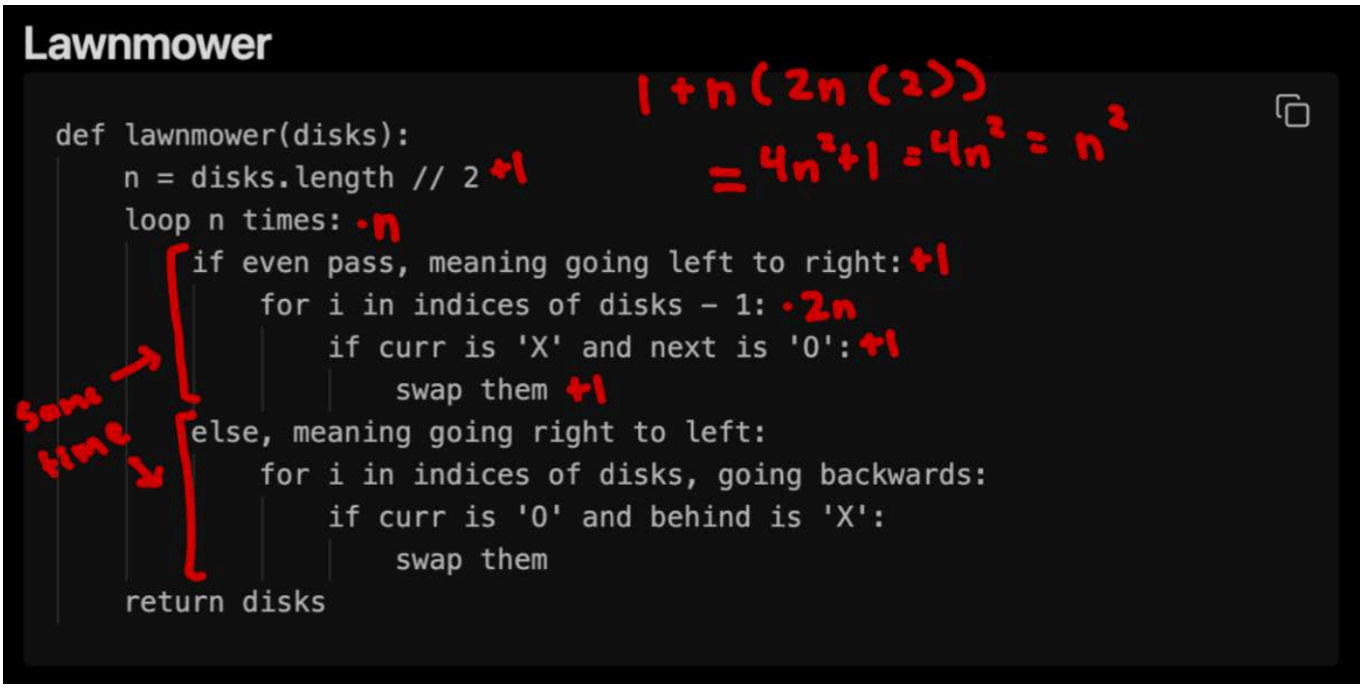
```
def left_to_right(disks):
    n = disks.length // 2
    loop n times:
        for i in indices of disks - 1:
            if curr is 'X' and next is 'O':
                swap them
    return disks
```



Lawnmower

```
def lawnmower(disks):
    n = disks.length // 2
    loop n times:
        if even pass, meaning going left to right:
            for i in indices of disks - 1:
                if curr is 'X' and next is 'O':
                    swap them
```

```
        else, meaning going right to left:
            for i in indices of disks, going backwards:
                if curr is 'O' and behind is 'X':
                    swap them
    return disks
```



The final step count for both of the algorithms is $4n^2$, so the time complexity for both is $O(n^2)$.