Marco Andrade

marcoandrade@csu.fullerton.edu

Dr. Sampson Akwafuo

CPSC 335(05) – Algorithm Engineering (Spring 2022)

19 February 2022

Project 1 Declaration

The Algorithm we are implementing:

- Lawnmower Algorithm: This Algorithm will start with the left-most disk and move to the right while swapping the disks when necessary. This Algorithm should put all the dark disks(0) to the left and all the light disks(1) to the right.
    - The user must input a positive integer (n)
    - The Algorithm will output

Implementing the pseudo-code:

- We assume that 0 will represent the dark disks and 1 will represent the light disks.

Pseudo code:

```
Define lawnmower(disk):
    swapCount = 0     //this will initially be zero, but will become 1 since the first disk is dark
    //first we want to scan the disks from left to right
    for I = 0 -> 2n do:  //take 2n-0+1 = 2n+1
        //we will start with the farthest disk to the left and move to the right
        for j = 0 -> 2n-1 do: //we are taking (2n-1)+1=2n
            //if the left is 0 and the right is 1. Move 0 to the left
            if(disk[j] is greater disk[j+1]):
                disk[j] = 0 //keep the dark disk on the left
                disk[j+1] = 1 //move the white disk to the right
                swapCount++ //increase the count by 1
    //now we want to scan the disks from right to left
    For k = 2n-1 -> 0 do:
```

```
If(disk[k] is less than disk[k-1]):
        disk[k] = 1
        disk[k-1] = 0
        swapCount++ //increase the count by 1
```

How are the steps counted?

- 1 + (2n+1)[(2n) x 6 + (2n) x 6] = 1 + (2n + 1) [24n^2] = 1 + 48n^2 + 24n = 48n^2 + 24n + 1
    - The time complexity for this calculation is O(n^2).

$$\lim_{n\to\infty}\left(\frac{48n^2+24n+1}{n\text{^}2}\right) = \lim_{n\to\infty}\left(\frac{96n+24}{2n}\right) =$$

$$\lim_{n\to\infty}\left(48+\frac{12}{n}\right) = 48 + 0 = 48 >= 0$$

$$48n^2 + 24n + 1 \in O(n^2)$$

What Could be better?

One thing that I would improve if I had more time would be implementing another algorithm. Something that would sort by taking the input from the user. Or I can simply make this Algorithm (Lawnmower algorithm) take the users' input instead of being hardcoded. I didn't implement it that way because I had many issues when I ran my program. If I had more time, I would've implemented it in that way since being hardcoded limits the number of moves the Algorithm can do.