CPSC 335 Fall 2019 MW 2:30-3:45 pm Project 1 Report

Carla Jacobsen cjacobsen2016@csu.fullerton.edu

Left-To-Right Algorithm:

```
assert(before.is_alternating())

unsigned numSwaps  = 0

//sc for entire loop:
while ! is_sorted do
        {//while
        for i = 0 to n do
                {//for
                if current disk is light and the next is dark then swap
                if (disk[i] == light) && (disk[i+1] == dark) do
                        {//if
                        ++numSwaps

                        swap disk[i] and disk[i+1]
                        }//if
                }//for
        }//while

return a new sorted_disks
```

Proof:

sc of swap = 13
sc of the if condition = 7 + 7 + 1 = 15
sc of the if condition if true = 15 + 1 + 13 = 29
sc of the full if statement = sc max(15, 29) = 29

sc of the for loop condition = n
sc of the entire for loop = 29n

sc of is_sorted function = 10n + 2
sc of while condition = n * (10n + 2) = 10(n^2) + 2n
sc of entire while loop contents = (10(n^2) + 2) * (29n) = 290(n^3) + 58(n^2)

sc of assert(before.is_alternating()) = 11n + 2
sc of numSwaps initialization = 1
sc of return statement = 2

sc of algorithm = 11n + 2 + 290(n^3) + 58(n^2) + 1 + 2 = 290(n^3) + 58(n^2) + 11n + 5 → O(n^3)

Lawnmower Algorithm:

```
assert(before.is_alternating())

unsigned numSwaps  = 0

//sc for entire loop:
while ! is_sorted do
        {//while
        for i = 0 to n do
                {//for
                if current disk is light and the next is dark then swap
                if (disk[i] == light) && (disk[i+1] == dark) do
                        {//if
                        ++numSwaps

                        swap disk[i] and disk[i+1]
                        }//if
                }//for
        for i = n-1 to 0 do
                {//for
                if current disk is dark and the left is light then swap
                if(disk[i] == dark) && (disk[i-1] == light) do
                        {//if
                        ++numSwaps

                        swap disk[i] and disk[i-1]
                        }//if
                }//for
        }//while

return a new sorted_disks //sc 2
```

Proof:

if statement in n for loop
sc of swap #1 = 13
sc of the if condition #1 = 7 + 7 + 1 = 15
sc of the if condition if true #1 = 15 + 1 + 13 = 29
sc of the full if statement #1 = sc max(15, 29) = 29

if statement in n - 1 for loop
sc of swap #2 = 13
sc of the if condition #2 = 7 + 7 + 1 = 15
sc of the if condition if true #2 = 15 + 1 + 13 = 29
sc of the full if statement #2 = sc max(15, 29) = 29

n for loop
sc of the for loop condition = n
sc of the entire for loop = 29n

n – 1 for loop
sc of the for loop condition = n – 1
sc of the entire for loop = 29n – 29

sc of both for loops = 29n + 29n – 29 = 58n - 29

sc of is_sorted function = 10n + 2
sc of while condition = (n/2) * (10n + 2) = 5(n^2) + n
sc of entire while loop contents = (5(n^2) + n) * (58n - 29) = 290(n^3) – 145(n^2) + 58(n^2) – 29n = 290(n^3) – 87(n^2) - 29n

sc of assert(before.is_alternating()) = 11n + 2
sc of numSwaps initialization = 1
sc of return statement = 2

sc of algorithm = 11n + 2 + 1 + 2 + 290(n^3) – 87(n^2) – 29n = 290(n^3) – 87(n^2) – 18n + 5 → O(n^3)