Submission for Project 1

Group members:

Jonathan Gaytan jagaytan@csu.fullerton.edu

Kyle Dang dangkyle@csu.fullerton.edu

Screenshots of Editor and Code Compiling and Executing:

Editor:

```
Activities
                 ◄ Visual Studio Code
                                                                                                 Mar 18 10:56 AM 🗓
                                                                                                                                                                                                 ▼ 🕬 🗐 100 9
                                                         README.md - project-lawnmover-Jonathan-Gaytan-and-Kyle-Dang - Visual Studio Code
File Edit Selection View Go Run Terminal Help
                                                                                                                                                                ф
                                             ① README.md > Ⅲ # project-lawnmover-Jonathan-Gaytan-and-Kyle-Dang
1 # project-lawnmover-Jonathan-Gaytan-and-Kyle-Dang
       V PROJECT-LAWNMOVER-JONATHAN-..
         gitignore
                                                    Group members:

    Jonathan Gaytan jagaytan@csu.fullerton.edu
    Kyle Dang dangkyle@csu.fullerton.edu

        M Makefile
                                                9 * input -> list of alternated colored disks, int n for amount of runs
10 * output -> list of organized disks
                                                     //BEFORE APPLICATION
                                                          for n to n + 1
    for position = 0 to rightmost_disk //EVEN
    if disk[2*position] is out_of_bounds
                                                                     break
else if disk[2*position - 1] is black
                                                                          switch disk[2*position] with disk[2*position-1]
                                                                     else
                                                               do nothing
for position = 1 to rightmost_disk-1 //ODD
if disk[2*position + 1] is out_of_bounds
                                                                     break
else if disk[2*position - 2] is black
       > OUTLINE
                                                                          switch disk[2*position +1] with disk[2*position -2]
```

Code Compiling and Executing:

Alternate Algorithm:

 $= 5 + (n + 1) * (4 + max(\frac{3n+6}{2}, \frac{n}{2} * 3))$

```
Pseudocode:
   for n to n + 1
     if even
        for position at 0 to rightmost disk
           if value of left disk > value of right disk
              swap position of disks
              add to numOfSwap
              position += 2
     else odd
        for position at 1 to rightmost disk - 1
           if value of left disk > value of right disk
              swap position of disks
              add to numOfSwap
              position += 2
   return sorted
Step Count:
s.\,c. = 1\,+\,1\,+\,2\,+\,1\,+\,(n\,-\,0\,+\,1)\,\,{}^*\,\left(2\,+\,2\,+\,max((\frac{n-0}{2}\,+\,1)\,\,{}^*\,(2\,+\,1),((\frac{(n-1)-1}{2}\,+\,1)\,\,{}^*\,(2\,+\,1))\right)
= 5 + (n + 1) * (4 + max((\frac{n}{2} + 1) * 3, (\frac{n-2}{2} + 1) * 3)
= 5 + (n + 1) * (4 + max((\frac{n+2}{2}) * 3, (\frac{n-2+2}{2}) * 3))
```

$$= 5 + (n + 1) * (4 + max(\frac{3n+6}{2}, \frac{3n}{2}))$$

$$= 5 + (n + 1) * (4 + \frac{3n+6}{2})$$

$$= 5 + (n + 1) * (\frac{3n+6+8}{2})$$

$$= 5 + (n + 1) * (\frac{3n+14}{2})$$

$$= 5 + (\frac{3n^2+14n}{2} + \frac{3n+14}{2})$$

$$= 5 + (\frac{3n^2+14n+3n+14}{2})$$

$$= 5 + (\frac{3n^2+17n+14}{2})$$

$$= \frac{3n^2+17n+14+10}{2}$$

$$= \frac{3n^2+17n+24}{2}$$

$$= \frac{3n^2+17n+24}{2}$$

Efficiency Class with Limit Theorem:

$$\lim_{n \to \infty} \frac{\frac{3}{2}n^2 + \frac{17}{2}n + 12}{n^2} = \frac{3}{2}$$

Due to $\frac{3}{2} \ge 0$ and $\frac{3}{2}$ being a constant, the Limit Theorem states that $\frac{3n^2 + 17n}{2} + 12 \in O(n^2)$.

That means this algorithm has a time complexity of $O(n^2)$.

Lawnmower Algorithm:

```
for n to n / 2

for position at 0 to rightmost_disk - 1

if value of left_disk > value of right_disk

swap the disks

add to numOfSwap

position++

for position at rightmost_disk - 2

if value of left_disk > value of right_disk

swap the disks

add to numOfSwap

position--
```

Step Count:

return sorted

$$s. c. = 1 + 1 + 1 + 1 + 1 + (n - 0 + 1) * (((n - 1) - 0 + 1) * (2 + 1) + (\frac{0 - (n - 2)}{-1} + 1) * (2 + 1))$$

$$= 4 + (n + 1) * (n * 3 + (\frac{-n + 2}{-1} + 1) * 3)$$

$$= 4 + (n + 1) * (3n + (n - 2 + 1) * 3)$$

$$= 4 + (n + 1) * (3n + (n - 1) * 3)$$

$$= 4 + (n + 1) * (3n + (3n - 3))$$

$$= 4 + (6n^2 - 3n + 6n - 3)$$

$$=6n^2+3n+1$$

Efficiency Class with Limit Theorem:

$$\lim_{n \to \infty} \frac{6n^2 + 3n + 1}{n^2} = 6$$

Due to $6 \ge 0$ and 6 being a constant, the Limit Theorem states that $6n^2 + 3n + 1 \in O(n^2)$. That means this algorithm has a time complexity of $O(n^2)$.