

```
student@tuffix-vm: ~/project-lawnmover
student@tuffix-vm:~/project-lawnmover$ g++ disks_test.cpp
student@tuffix-vm:~/project-lawnmover$ ./a.out
disk_state still works: passed, score 1/1
sorted_disks still works: passed, score 1/1
disk_state::is_initialized: passed, score 3/3
disk_state::is_sorted: passed, score 3/3
alternate, n=4: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 14 / 14

student@tuffix-vm:~/project-lawnmover$
```

Alternate Sorting Method

Pseudocode:

```
/*
int numOfSwap = 0;
disk_state state = before;
for int i = 0 to state's total count + 1 do
    if i % 2 == 0 do
        for int index = 0 to total count - 1 so that index = index + 2 do
            if current disk != next disk
                if current disk == dark disk && next disk == light disk
                    swap both disks
                numOfSwap++;
```

```

        end if
    end if
end for loop
end if
else
    for int index = 1 to total count - 2 so that index = index + 2 do
        if current disk != next disk do
            if current disk == dark disk && next disk == light disk
                swap both disks
                numOfSwap++;
            end if
        end if
    end for loop
end else
return the sorted disks
*/

```

Step Count:

```

int numOfSwap = 0; (1)
disk_state state = before; (1)
for int i = 0 to state's total count + 1 do ( $n + 1 + 1 = n + 2$ )
    if  $i \% 2 == 0$  do ( $2 + \text{max of...}$ )
        for int index = 0 to total count - 1 so that index = index + 2 do (n)
            if current disk != next disk ( $1 + \text{max of...}$ )
                if current disk == dark disk && next disk == light disk ( $2 + \text{max of...}$ )
                    swap both disks (0)
                    numOfSwap++; (0)
                end if
            end if
        end for loop
    end if
end if

```

```

else
  for int index = 1 to total count - 2 so that index = index + 2 do (n - 2)
    if current disk != next disk do (1 + max of...)
      if current disk == dark disk && next disk == light disk (2 + max of...)
        swap both disks (0)
        numOfSwap++; (0)
      end if
    end if
  end for loop
end else
return the sorted disks (0)

```

$$\begin{aligned}
 \text{Step Count} &= 1 + 1 + (n + 2)(2 + \max(n + 1 + \max(2 + \max(0)), n - 2 + 1 + \max(2 + \max(0)))) \\
 &= 2 + (n + 2)(2 + \max(n + 1 + \max(2), n - 1 + \max(2))) \\
 &= 2 + (n + 2)(2 + \max(n + 3, n + 1)) \\
 &= 2 + (n + 2)(n + 5) \\
 &= n^2 + 7n + 10 + 2 = n^2 + 7n + 12
 \end{aligned}$$

Time Complexity:

$$n^2 + 7n + 12 \text{ in } O(n^2)?$$

$$n^2 + 7n + 12 \leq c * n^2, n \geq n_0$$

$$\text{Let } c = 1 + 7 + 12 = 20 \text{ and } n_0 = 1$$

$$n^2 + 7n + 12 \leq 20n^2, n \geq 1$$

This is trivially true, therefore $n^2 + 7n + 12$ does have a time complexity of $O(n^2)$.

Lawnmover's Algorithm:

Pseudocode:

```
/*
```

```
  int numOfSwap = 0;
```

```
  disk_state state = before;
```

```

for int i = 0 to all lights do
  int index = 0;
  while index + 1 < n do
    if current disk != next disk do
      if current disk == dark disk && next disk == light disk do
        swap disks
        numOfSwap++;
      end if
    end if
    index++;
  end while
  while index > 0 do
    if previous disk != current disk do
      if previous disk == dark disk && current disk == light disk do
        swap disks
        numOfSwap++;
      end if
    end if
    index--;
  end while
  return sorted disks
*/

```

Step Count:

```

int numOfSwap = 0; (1)
disk_state state = before; (1)
for int i = 0 to n / 2 do ((n/2) + 1)
  int index = 0; (1)
  while index + 1 < n do (n - 1)
    if current disk != next disk do (1 + max of...)
      if current disk == dark disk && next disk == light disk do (2 + max of...)

```

```

    swap disks (0)
    numOfSwap++; (0)
end if
end if
index++; (0)
end while
while index > 0 do (n)
    if previous disk != current disk do (1 + max of...)
        if previous disk == dark disk && current disk == light disk do (2 + max of...)
            swap disks (0)
            numOfSwap++; (0)
        end if
    end if
    index--; (0)
end while
return sorted disks (0)

```

$$\begin{aligned}
 SC &= 1 + 1 + ((n/2) + 1)(1 + n - 1 + 1 + \max(2 + \max(0)) + 0 + n + 1 + \max(2 + \max(0)) + 0) \\
 &= 2 + ((n/2) + 1)(n + 1 + \max(2) + n + 1 + \max(2)) \\
 &= 2 + ((n/2) + 1)(2n + 6) \\
 &= 2 + n^2 + 5n + 6 = n^2 + 5n + 8 \\
 SC &= n^2 + 5n + 8
 \end{aligned}$$

Time Complexity:

$$n^2 + 5n + 8 \text{ in } O(n^2)?$$

$$n^2 + 5n + 8 \leq c * n^2, n \geq n_0$$

$$\text{Let } c = 1 + 5 + 8 = 14 \text{ and } n_0 = 1$$

$$n^2 + 5n + 8 \leq 14n^2, n \geq 1$$

This is trivially true, therefore $n^2 + 5n + 8$ does have a time complexity of $O(n^2)$.