

# **Programming Assignment #1**

## **Table of Contents**

<b>Pseudocode .....</b>	<b>2</b>
<b>Big Oh Efficiency .....</b>	<b>2</b>
<b>Left To Right Algorithm C++ Code .....</b>	<b>4</b>
<b>Lawnmower Algorithm C++ Code .....</b>	<b>5</b>
<b>Sample Output .....</b>	<b>6</b>

# Pseudocode/Big Oh Efficiency

## Left to right sorting problem:

input: n, List “disks” of 2n disk alternating colors dark-dark-light-light

output: List L where dark disks are to far left and light disks are to far right (dark-dark-dark-dark-light-light-light-light)

```
def left_To_Right(disks):
    if |disks| = 0:
        return None
    else:
        moves = 0
        for k = 0 to n do
            for j = 0 to 2n-1 do
                if disks[j] greater than disks[j+1] do:
                    disks[j] = 0
                    disks[j+1] = 1
                    increment moves

        return sorted list disks
```

## Big Oh Efficiency:

```

moves = 0      1
for k = 0 to 2*n do
    Inner Loop A    for j = 0 to 2n-1 do    -> [(2n-1 - 0) / 1] + 1 = 2n
                    if disks[j] greater than disks[j+1] do: 1 + max(3, 0) = 4
                        disks[j] = 0                1
                        disks[j+1] = 1              1      - 3 total units
                        increment moves              1
Total number of steps in Loop A: 1 + (2n * 4) = 8n + 1

```

$$\sum_{k=0}^{2N-1} 8n + 1 = 8n + \sum_{k=0}^{2N-1} 1$$

$$= 8n(2n+1) + 2n+1 = (16n^2 + 8n) + 2n + 1 = 16n^2 + 10n + 1$$

**Time Complexity:  $O(n^2)$**

**Lawnmower sorting problem:**

input: n, List “disks” of 2n disk alternating colors dark-dark-light-light

output: List L where dark disks are to far left and light disks are to far right (dark-dark-dark-dark-light-light-light-light)

```
def lawnmower_Sort(disks):
    if |disks| = 0:
        return None
    else:
        moves = 0
        for k = 0 to 2*n do
            for j = 0 to 2n-1 do
                if disks[j] greater than disks[j+1] do:
                    disks[j] = 0
                    disks[j+1] = 1
                    increment moves
            //Going right to left
            for t = 2n-1 to 1 do
                if disks[t] is less than disks[t-1] do
                    disks[t] = 1
                    disks[t-1] = 0
                    increment moves

        return sorted list disks
```

**Big Oh Efficiency:**

moves = 0      1

**Outer Loop C** for k = 0 to 2\*n do

**Inner Loop A**      for j = 0 to 2n-1 do       $\rightarrow [(2n-1 - 0) / 1] + 1 = 2n$

                                 if disks[j] greater than disks[j+1] do: 1 + max(3, 0) = 4

   disks[j] = 0      1

   disks[j+1] = 1      1      - 3 total units

   increment moves      1

**Total number of steps in Loop A:  $(2n * 4) = 8n$**

**Inner Loop B**      for t = 2n-1 to 1 do       $\rightarrow |(1 - 2n - 1)/1| + 1 = 2n$

                                 if disks[t] is less than disks[t-1] do      1 + max(3, 0) = 4

   disks[t] = 1      1

   disks[t-1] = 0      1

   increment moves      1

**Total number of steps in Loop B:  $(2n - 1) * 4 = 8n$**

**Total number of steps in A + B = 16n**

$$\sum_{k=0}^{2N} 16n = 16n(2n+1) = 32n^2 + 16n + 1 \quad \text{Time Complexity: } O(n^2)$$

# Left To Right Algorithm C++ Code

```
void print_disks(int n, char *disks)
{
    cout << "List of disks\n";
    for (int i = 0; i < 2*n; ++i)
    {
        if (disks[i] == 0)
        {
            cout << "D ";
        }
        else
        {
            cout << "L ";
        }
    }
    cout << endl;
}

// reset the number of moves to 0;
m = 0;

// loop to push dark ones before light ones
for (k=0; k < 2*n ; k++)
{
    for (int j = 0; j < 2*n - 1; j++)
    {
        if (disks[j] > disks[j+1])
        {
            disks[j] = 0;
            disks[j+1] = 1;
            m++;
        }
    }
}
```

# Lawnmower Algorithm C++ Code

```
// reset the number of moves to 0;
m = 0;
// loop to push light one before darks ones

for (k=0; k < 2*n ; k++)
{
    //Going left to right
    for (int j = 0; j < 2*n - 1; j++)
    {
        if (disks[j] > disks[j+1])
        {
            disks[j] = 0;
            disks[j+1] = 1;
            m++;
        }
    }

    //Going right to left
    for (int t = 2*n - 1; t > 0; t--)
    {
        if (disks[t] < disks[t-1])
        {
            disks[t] = 1;
            disks[t-1] = 0;
            m++;
        }
    }
}
```

# Sample Output

## Left To Right Algorithm

```
Activities Terminal Thu 21:09
me@tla-ubuntu-gnome: ~/Desktop

File Edit View Search Terminal Help
me@tla-ubuntu-gnome:~$ cd Desktop/
me@tla-ubuntu-gnome:~/Desktop$ g++ LeftToRight.cpp
me@tla-ubuntu-gnome:~/Desktop$ ./a.out

CPSC 335-x - Programming Assignment #1
The alternating disks problem: left-to-right algorithm
Enter an even number of single color disks (light or dark)
4
Initial configuration
List of disks
D D L L D D L L
After moving darker ones to the left
List of disks
D D D D L L L L
Number of swaps is 4
me@tla-ubuntu-gnome:~/Desktop$ ./a.out

CPSC 335-x - Programming Assignment #1
The alternating disks problem: left-to-right algorithm
Enter an even number of single color disks (light or dark)
6
Initial configuration
List of disks
D D L L D D L L D D L L
After moving darker ones to the left
List of disks
D D D D D D L L L L L L
Number of swaps is 12
me@tla-ubuntu-gnome:~/Desktop$
```

## Lawnmower Algorithm

```
Activities Terminal Thu 21:10
me@tla-ubuntu-gnome: ~/Desktop

File Edit View Search Terminal Help
me@tla-ubuntu-gnome:~$ cd Desktop/
me@tla-ubuntu-gnome:~/Desktop$ g++ Lawnmower.cpp
me@tla-ubuntu-gnome:~/Desktop$ ./a.out

CPSC 335-x - Programming Assignment #1
The alternating disks problem: lawnmower algorithm
Enter the number of single color disks (light or dark)
4
Initial configuration
List of disks
D D L L D D L L
After moving darker ones to the left
List of disks
D D D D L L L L
Number of swaps is 4
me@tla-ubuntu-gnome:~/Desktop$ ./a.out

CPSC 335-x - Programming Assignment #1
The alternating disks problem: lawnmower algorithm
Enter the number of single color disks (light or dark)
6
Initial configuration
List of disks
D D L L D D L L D D L L
After moving darker ones to the left
List of disks
D D D D D D L L L L L L
Number of swaps is 12
me@tla-ubuntu-gnome:~/Desktop$
```