PROGRAMMING ASSIGNMENT #1

Student Name: **Phuc Le**

Course: CPSC 335

Section: 1

- INPUT: an even integer n and a list of 2n disks of alternating colors light-dark, starting with light.
- OUTPUT: a list of 2n disks, the first n disks are dark and the next n disks are light, and an integer swapNumber representing the number of moves necessary to move the dark ones before the light ones

Pseudo Code for Left-to-Right Algorithm

```
def move_disks(n, disks):                          Time units
    swapNumber = 0                                 1
    for(i=0 to n) do:                              n + 1
        for k=i to (2n-i) step 2 do:               n - i + 1
            tempValue = disks[k]                   1   }
            disks[k] = disks[k+1]                  2   }
            disks[k+1] = tempValue                 2   }  7
            increment swapNumber by 1              2   }
        end for
    end for
    print disks                                    1
    print swapNumber                               1
```

* Running Time: $\frac{7}{2}n^2 + \frac{21}{2}n + 10$

in details: Running time $= 3 + \sum_{i=0}^{n} 7(n-i+1)$

$$= 3 + 7\sum_{i=0}^{n} n - 7\sum_{i=0}^{n} i + 7\sum_{i=0}^{n} 1$$

$$= 3 + 7n(n+1) - 7\frac{n(n+1)}{2} + 7(n+1)$$

$$= 3 + 7n^2 + 7n - \frac{7}{2}n^2 - \frac{7n}{2} + 7n + 7 = \frac{7}{2}n^2 + \frac{21}{2}n + 10$$

Left-to-Right Algorithm

* Prove Time Complexity using definition

$f(n) = \frac{7}{2}n^2 + \frac{21}{2}n + 10$

$g(n) = n^2$

$f(n) \in O(g(n))$ $\exists c > 0,\ n_0 \geq 0$

$c = ?,\ n_0 = ?,\ f(n) \leq c \cdot g(n)\ \forall\ n \geq n_0$

pick $c = 25$

$$\frac{7}{2}n^2 + \frac{21}{2}n + 10 \leq 25n^2$$

$$n \geq 1$$

Thus, with $c = 25$, $n_0 = 1$, the relationship is TRUE

* Prove Time Complexity using Limit

$$L = \lim_{n \to \infty} \frac{\frac{7}{2}n^2 + \frac{21}{2}n + 10}{n^2} = \lim_{n \to \infty} \frac{7}{2} + \frac{21}{2n} + \frac{10}{n^2} = \frac{7}{2}$$

because L is a non-negative constant, the relationship

$f(n) \in O(g(n))$ is TRUE

Pseudo Code for Lawnmower Algorithm

```
                                                          Time Units
def move_disks(n, disks)
    swapNumber = 0                                        1
    for i=0 to n do:                                      n+1
        for k=(2n-i-1) to i step (-2) do:                 n-i-1/2
            tempValue = disks[k]                          1 }
            disks[k] = disks[k-1]                         2 }
            disks[k-1] = tempValue                        2 } 7
            increase swapNumber by 1                      2 }
        end for
    end for
    print disks                                           1
    print swapNumber                                      1
```

* Running Time : $\frac{7}{2}n^2 - \frac{1}{2}$

in details : $\text{Running Time} = 3 + \sum_{i=0}^{n}\left(\frac{i-(2n-i-1)}{-2} + 1\right) \times 7$

$= 3 + 7\sum_{i=0}^{n}(n - i - \frac{1}{2}) = 3 + 7\sum_{i=0}^{n} n - 7\sum_{i=0}^{n} i - \frac{7}{2}\sum_{i=0}^{n}$

$= 3 + 7n(n+1) - 7\frac{n(n+1)}{2} - \frac{7}{2}(n+1)$

$= 3 + 7n^2 + 7n - \frac{7}{2}n^2 - \frac{7}{2}n - \frac{7}{2}n - \frac{7}{2}$

$= \frac{7}{2}n^2 - \frac{1}{2}$

Lawnmower Algorithm

* Prove the Time Complexity using definition

$f(n) = \frac{7}{2}n^2 - \frac{1}{2}$

$g(n) = n^2$

$f(n) \in O(g(n)) \quad \exists c > 0, n_0 \geq 0$

$c = ?, n_0 = ?, f(n) \leq c \cdot g(n) \quad \forall n \geq n_0$

pick $c = \frac{3}{2}$

$\frac{7}{2}n^2 - \frac{1}{2} \leq \frac{3}{2}n^2$

and with $n_0 = \frac{1}{2}$ the relationship $f(n) \in O(g(n))$ is TRUE

* Prove Time Complexity using Limit

$$L = \lim_{n \to \infty} \frac{\frac{7}{2}n^2 - \frac{1}{2}}{n^2} = \lim_{n \to \infty} \left(\frac{7}{2} - \frac{1}{2n^2}\right) = \frac{7}{2}$$

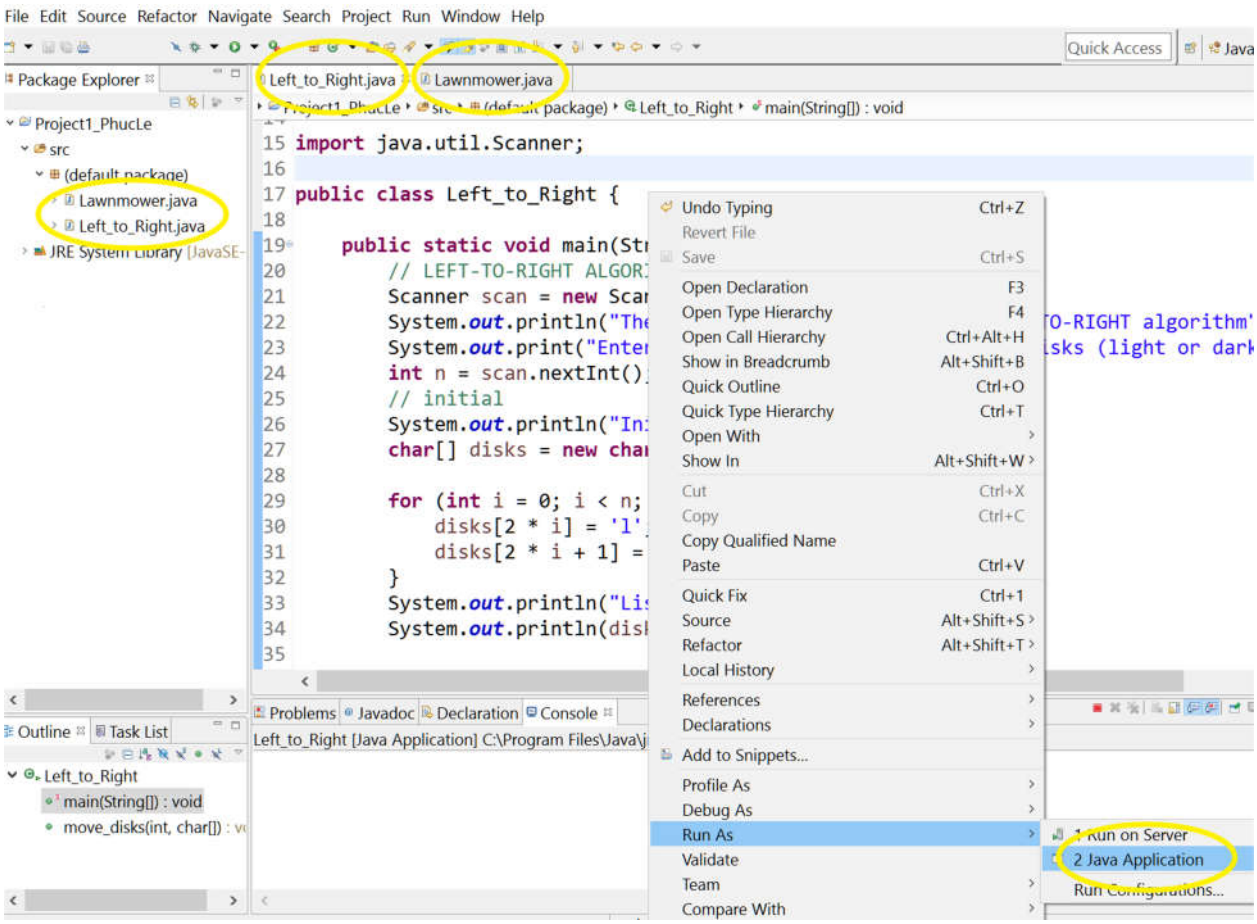
Because L is a non-negative constant, the relationship $f(n) \in O(g(n))$ is TRUE

* Number of swaps: for both algorithm, the number of swaps

$$= \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

How to Run the Source Code - <u>Method 1</u>:

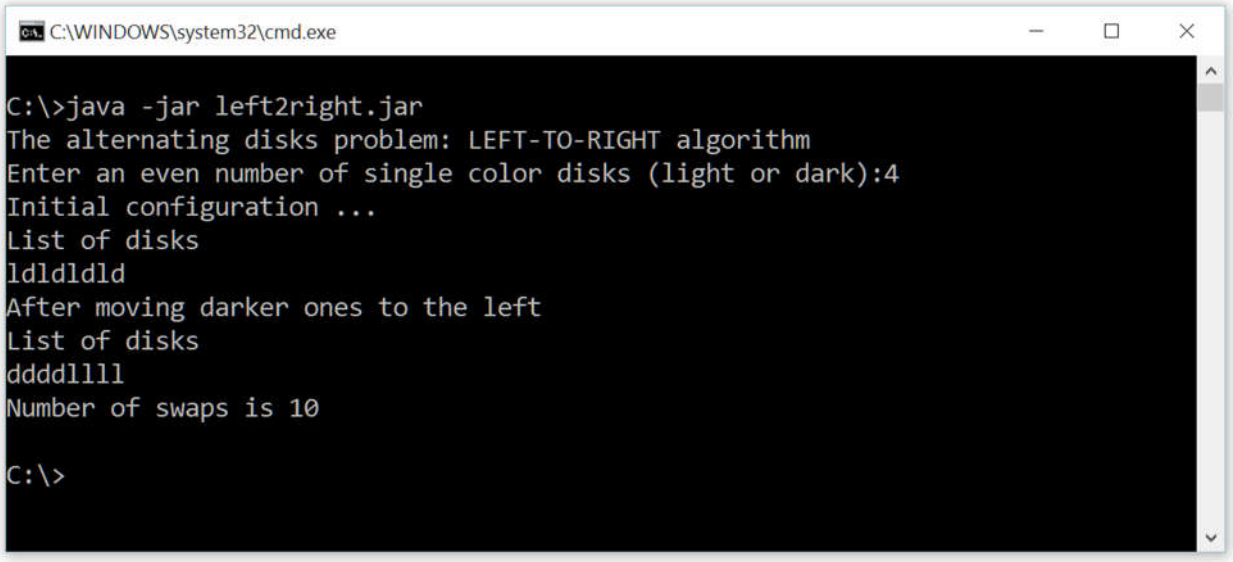
import project into Eclipse Mar (version 4.5), open the appropriate algorithm file, right click on the source code, click “Run As”, and click “Java Application”



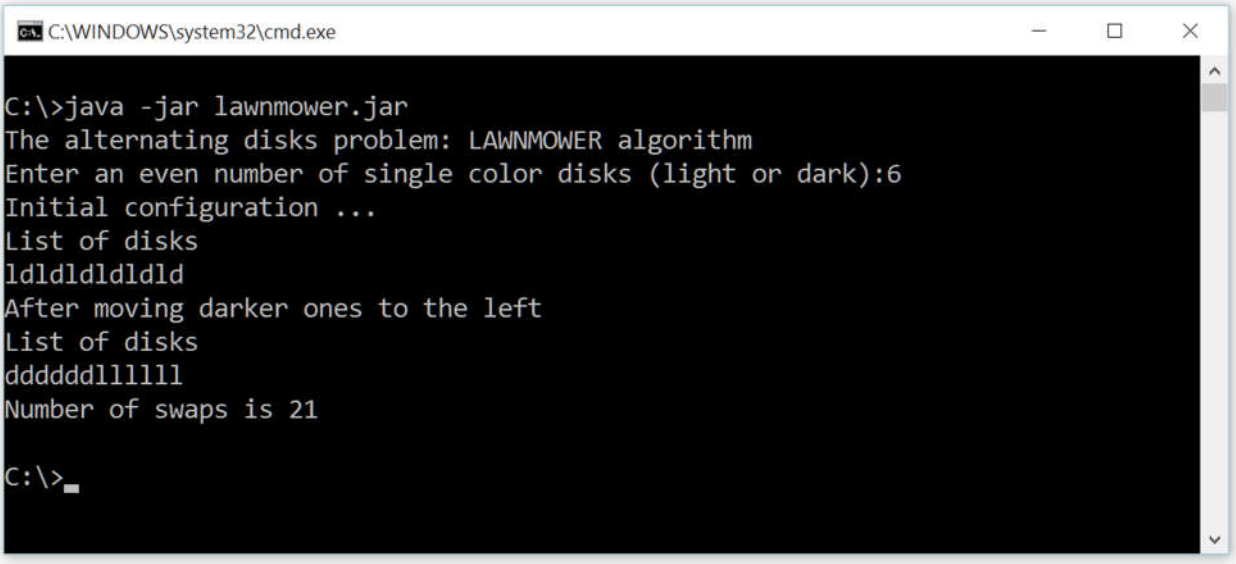
How to Run the Source Code - <u>Method 2</u>:

copy 2 files: "***left2right.jar***" and "***lawnmower.jar***" inside folder "***Executable Files***" into ***C:*** drive, open the console windows of commands and type the commands as the demonstration below:

Java –jar left2right.jar



Java –jar lawnmower.jar



The Output Examples:

Example 1:

```
The alternating disks problem: LEFT-TO-RIGHT algorithm
Enter an even number of single color disks (light or dark):8
Initial configuration ...
List of disks
ldldldldldldldld
After moving darker ones to the left
List of disks
ddddddddllllllll
Number of swaps is 36
```

Example 2:

```
The alternating disks problem: LAWNMOWER algorithm
Enter an even number of single color disks (light or dark):6
Initial configuration ...
List of disks
ldldldldldld
After moving darker ones to the left
List of disks
dddddd llllll
Number of swaps is 21
```

Example 3:

```
The alternating disks problem: LAWNMOWER algorithm
Enter an even number of single color disks (light or dark):4
Initial configuration ...
List of disks
ldldldld
After moving darker ones to the left
List of disks
ddddllll
Number of swaps is 10
```