Team members:
Cong Le Cle155@csu.fullerton.edu
Jack Nguyen nhannguyen7@csu.fullerton.edu
CPSC335-03: Algorithm Engineering
Professor Kevin Wortman
Due date: 5 Oct, 2018

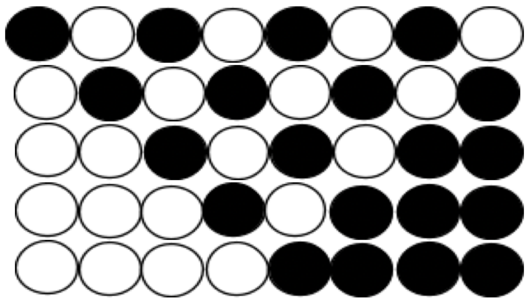
Project 1- Implementing Algorithms

The alternating disks problem: **Input**: a positive integer n and a list of 2n disks of alternating colors dark-light, starting with dark **Output**: a list of 2n disks, the first n disks are light, the next n disks are dark, and an integer m representing the number of swaps to move the dark ones after the light ones.

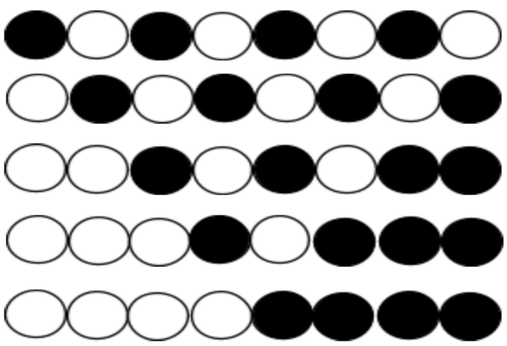
The following two algorithms (Left-to-Right and Lawnmower) will solve this problem in $O(n^2)$ time. We will analyze the two algorithms in details by examine their pseudocodes and time complexity.

<u>Part I</u>: Description of the algorithms:

1. **Left to right algorithm:** It starts with the leftmost disk and proceeds to the right, doing the swaps as necessary. Now we have one lighter disk at the left-hand end and the darker disk at the right-hand end. Once it reaches the right-hand end, it goes back to the leftmost disk and proceeds to the right, doing the swaps as necessary. It repeats until there are no more disks to move.



2. **Lawnmower algorithm:** It starts with the leftmost disk and proceeds to the right, doing the swaps as necessary. Now we have one lighter disk at the left-hand end and the darker disk at the right-hand end. Once it reaches the right-hand end, it starts with the disk before the rightmost disk and proceeds to the left, doing the swaps as necessary, until it reaches the disk before the left-hand end. It repeats until there are no more disks to move.



Part II: Pseudocode for the algorithms and step count:

1. **Left to right algorithm:**

*Define number of swaps **m*** 1
*For **i** from **0** through **n** in the unsorted array **A**:* 1
*For **j** from **j = i** through **2n-1** in the unsorted array **A**:* 1
*If **A[j]** > A[**j+1**]* 1
*Swap (**A[j]**, **A[j+1]**)* 1 2n-1 n
*Increment **m*** 1

*Return (array **A**, **m**)* 1

2. **Lawnmower algorithm:**

*Define number of swaps **m*** 1
*For **i** from **0** through **n** in the unsorted array **A**:* 1
*For **j** from **j = i** through **2n-1** in the unsorted array **A**:* 1
*If **A[j]** > A[**j+1**]* 1
*Swap (**A[j]**, **A[j+1]**)* 1 2n-1
*Increment **m*** 1 n
*For **j** from **j = 2n-1** to **j = 0** through in the unsorted array **A**:* 1
*If **A[j]** < A[**j-1**]* 1
*Swap (**A[j]**, **A[j-1]**)* 1 2n-1
*Increment **m*** 1
*Return (array **A**, **m**)* 1

Part III: Mathematical Analysis for the algorithms:

1. **Left-to-Right algorithm:**

$$\begin{aligned} T(n) &= 1 + n[1 + (2n-1)(1+1+1+1)] + 1 \\ &= 2 + n[1 + (2n-1)(4)] \\ &= 2 + n + 4n(2n-1) \\ &= 2 + n + 8n^2 - 4n \\ &= 8n^2 - 3n + 2 \\ &\in O(8n^2 - 3n + 2) \quad \text{(trivial)} \\ &= O(8n^2) \quad \text{(dominated term)} \\ &= O(n^2) \quad \text{(constant factor)} \end{aligned}$$

$$\therefore \textit{The Left-to-Right algorithm takes } O(n^2) \textit{time}$$

2. **Lawnmower algorithm:**

$$\begin{aligned} T(n) &= 1 + n[1 + (2n-1)(1+1+1+1) + (2n-1)(1+1+1+1)] + 1 \\ &= 2 + n[1 + (2n-1)(4) + (2n-1)(4)] \\ &= 2 + n + 4n(2n-1) + 4n(2n-1) \\ &= 2 + n + 8n^2 - 4n + 8n^2 - 4n \\ &= 16n^2 - 7n + 2 \\ &\in O(16n^2 - 7n + 2) \quad \text{(trivial)} \\ &= O(16n^2) \quad \text{(dominated term)} \\ &= O(n^2) \quad \text{(constant factor)} \end{aligned}$$

$$\therefore \textit{The Lawnmower algorithm takes } O(n^2) \textit{time}$$

Part IV: Tuffix Environment Proof