John Hunsaker

CPSC 335

Algorithm Engineering

ALTERNATING DISKS PROJECT

I. Introduction

The goal of this paper is to design, implement, and analyze two straightforward greedy algorithms for the alternating disks problem. The following steps will be covered throughout this paper for each algorithm:

A. Describe the design of a greedy based method for each algorithm and represent it with pseudocode.

B. Analyze and mathematically prove the efficiency of the pseudocode using Big-Oh notation.

C. Source Code Implementation of this pseudocode using Java.

D. Test cases for various input sizes and output for each input.

II. Alternating Disks Problem

The problem, presented in Levitin's textbook as Exercise 14 on page 103, is as follows:

You have a row of 2n disks of two colors, n dark and n light. They alternate: dark, light, dark, light, and so on. You want to get all the dark disks to the right-hand end, and all the light disks to the left-hand end. The only moves you are allowed to make are those that interchange the positions of two neighboring disks. Design an algorithm for solving this puzzle and determine the number of moves it takes.

The *alternating disks problem* is:

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

III. Left-to-Right Algorithm

This algorithm starts with the leftmost disk and proceeds to the right, doing the swaps as necessary. Now there is 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.

A. Pseudocode

```
ALGORITHM Left-to-right
// Input: a positive integer n and a list of 2n disks of alternating colors within an array A[n]
where dark being represented as 1's and light represented as 0's
// Output: a list of 2n disks, the first n disks are light, the next n disks are dark in the form
of an array A[n] where dark is being represented as 1's and light is represented as 0's

Number        ← 0                                   // Number of disks to be inputted by user
swaps         ← 0
rightMostDisk ← (2 * number) – 2                    // Track the rightmost disk's index

For i = 0 to number do
      For a = 0 to rightMostDisk do
            if( A[a] == 'B' && A[a+1] == 'W')       // if left is dark and right is light
                  swap A[a] with A[a+1]             // swap them
                  swaps++                           // increment Swap count
            end if
      end for
end for
```

B. Efficiency

We will now calculate the efficiency of the algorithm based of pseudocode's basic operation defined above:

$$C(n) = \sum_{i=0}^{n} \sum_{j=0}^{2n-2} 1 \qquad [Eq. 1]$$

Using the rules defined in the textbook (R1), (R2), (S1), and (S2) we can simply the summation above to:

$$C(n) = \sum_{i=0}^{n} \sum_{j=0}^{2n-2} 1 = \sum_{i=0}^{n} [(2n-2) - 0 + 1] = \sum_{i=0}^{n} 2n - 1 = 2n \sum_{i=0}^{n} 1 - \sum_{i=0}^{n} 1 = [2n(n+1)] - (n+1)$$

$$C(n) = 2n^2 + n - 1$$

B.1. Proof by Mathematical definition

$$2n^2 + n - 1 \, \epsilon \, O(n^2) \; if \; 2n^2 + n - 1 \leq cn^2 \; for \; all \; n \geq n_0$$

$$2n^2 + n - 1 \epsilon \, O(2n^2 + n - 1) \qquad \rightarrow [Trivial]$$

$$\leq O(\max(2n^2, n, 1)) \qquad \rightarrow [Drop \; dominated \; terms]$$

$$\leq O(2n^2)$$

$$\leq O(n^2) \qquad \rightarrow [Drop \; multiplicative \; constants]$$

$$\therefore By \; definition \; 2n^2 + n - 1 \, \epsilon \, O(n^2)$$

B.2. Proof by Limits Theorem

$$\lim_{n \to \infty} \frac{2n^2 + n - 1}{n^2} = \lim_{n \to \infty} \frac{4n + 1}{2n} = 3 \geq 0 \qquad \rightarrow [constant]$$

$$\therefore By \; limits \; theorem, 2n^2 + n - 1 \epsilon \, O(n^2)$$

IV. Lawnmower Algorithm

This algorithm starts with the leftmost disk and proceeds to the right, doing the swaps as necessary. While near the right-hand end, it reverses direction i.e. moving right to left now, performing swaps as necessary as it continues. It repeats until all of the dark discs are on the right-half, and all the light discs are on the left-half.

A. Pseudocode

```
ALGORITHM Lawnmower
// Input: a positive integer n and a list of 2n disks of alternating colors within an array A[n]
where dark being represented as 1's and light represented as 0's
// Output: a list of 2n disks, the first n disks are light, the next n disks are dark in the form
of an array A[n] where dark is being represented as 1's and light is represented as 0's

number        ← 0                                   //Number of disks to be inputted by user
swaps         ← 0
storeVal      ← 0
rightMostDisk ← (2 * number) – 2                    // Keeps track of the rightmost disk's index


for i = 0 to (number/2) do
      for a = i to rightMostDisk
            if( A[a] == 'B' && A[a+1] == 'W')       // if left is dark and right is light
                  swap A[a] with A[a+1]             // swap them
                  swaps++                           // increment swap count
                  storeVal = a;
            end if
      end for

      for a = storeVal to i, decrement i
            if( A[a] == 'B' && A[a+1] == 'W')
                  swap A[a] with A[a+1]             // swap them
                  swaps++                           // increment swap count
            end if
      end for
end for
```

B. Efficiency

We will now calculate the efficiency of the algorithm based of pseudocode's basic operation defined above:

$$C(n) = \sum_{i=0}^{\frac{n}{2}+1} 1 \left(\sum_{j=i}^{2n-2} 1 + \sum_{k=j}^{i} 1 \right) \qquad [Eq.\,2]$$

First, we solve the summations inside the parenthesis:

$$\sum_{j=i}^{2n-2} 1 = (2n-2) - i + 1 = (2n-1-i) \qquad [Eq.\,2.1]$$

$$\sum_{k=j}^{i} 1 = (i-j+1) \qquad [Eq.\,2.2]$$

$$(2n-1-i) + (i-j+1) = 2n-j \qquad [Eq\ 2.3]$$

After solving the summations inside the parenthesis, we apply must use the rules in the book to expand the summation:

$$= \sum_{i=0}^{\frac{n}{2}+1} 2n + \sum_{i=0}^{\frac{n}{2}+1} -j \qquad [Eq.2.4]$$

We will solve each summation separately and sum them after they have been simplified:

$$\rightarrow \sum_{i=0}^{\frac{n}{2}+1} 2n = 2\mathrm{n} \sum_{i=0}^{\frac{n}{2}+1} 1 = 2\mathrm{n}\left(\frac{\mathrm{n}}{2} - 0 + 1\right) = \mathrm{n}(\mathrm{n}+4) \qquad [Eq.2.5]$$

$$\rightarrow \sum_{i=0}^{\frac{n}{2}+1} -j = -\sum_{i=0}^{\frac{n}{2}+1} j = -\frac{\frac{n}{2}\left(\frac{n}{2}+1\right)}{2} = -\frac{n(n+2)}{8} \qquad [Eq.2.6]$$

$$= [\mathrm{n}(\mathrm{n}+4)] - \frac{n(n+2)}{8} = \frac{7n^2}{8} + \frac{15n}{4} \qquad [Eq.2.7]$$

B.1. Proof by Mathematical definition

$$\frac{7n^2}{8} + \frac{15n}{4} \epsilon\, O(n^2)\ if\ \frac{7n^2}{8} + \frac{15n}{4} \leq cn^2\ for\ all\ n \geq n_0$$

$$\frac{7n^2}{8} + \frac{15n}{4}\ \epsilon\, O\left(\frac{7n^2}{8} + \frac{15n}{4}\right) \qquad \rightarrow [Trivial]$$

$$\leq O\left(\max\left(\frac{7n^2}{8}, \frac{15n}{4}\right)\right) \qquad \rightarrow [Drop\ dominated\ terms]$$

$$\leq O\left(\frac{7n^2}{8}\right)$$

$$\leq O(n^2) \qquad \rightarrow [Drop\ multiplicative\ constants]$$

$$\therefore By\ definition, \frac{7n^2}{8} + \frac{15n}{4} \epsilon\, O(n^2)$$

B.2. Proof by Limits Theorem

$$\lim_{n\to\infty} \frac{\frac{7n^2}{8}+\frac{15n}{4}}{n^2} = \lim_{n\to\infty} \frac{\frac{49n}{8}+\frac{15}{4}}{2n} = \frac{49}{16}+0 = \frac{49}{16} \geq 0 \qquad \rightarrow [constant]$$

$$\therefore By\ limits\ theorem, \frac{7n^2}{8}+\frac{15n}{4}\ \epsilon\ O(n^2)$$