CPSC 335 Project 1

Group members / email address

Sammy Guzman: sammyguzman@csu.fullerton.edu

Hyojun Lee : lee.h@csu.fullerton.edu

```
student@tuffix-vm:/media/sf_Downloads/prj1-lawnmower$ make
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
./disks_test
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:/media/sf_Downloads/prj1-lawnmower$ S
```

The lawnmower algorithm

Algorithm Design:

Pseudo Code

end for

Input: a positive integer **n** and a list of **2n** disks of alternating colors -light-disk Output: List of 2n disks, m for the number of swaps

```
for i = 0 to n/2 do
   for j = 0 to 2n-1 do
      if list[j] > list[j+1]
         swap
      end
   end for
    for j = 2n-1 to 0 do
      if list[j] < list[j-1]</pre>
   swap
      end
   end for
end for
Step Count
                             // SC outer loop =
for i = 0 to n/2 do
                                // (n+1)/2 times
                               // SC forward loop
   for j = 0 to 2n-1 do
                                // 2n times
      if list[j] > list[j+1]
                                // 1 TU
         Swap
                                 // 3 TU
        m++
                                 // 1 TU
      end
   end for
                              // SC reverse loop
  for j = 2n-1 to 0 do // 2n times
      if list[j] < list[j-1]</pre>
                                // 1 TU
                                 // 3 TU
        Swap
                                 // 1 TU
        M++
      end
   end for
```

$$S. C_{forward} = 2n(1 + max(4, 0))$$

$$= 2n(1 + 4)$$

$$= 2n(5)$$

$$= 10n$$

$$S. C_{reverse} = 2n(1 + max(4, 0))$$

$$= 2n(1 + 4)$$

$$= 2n(5)$$

$$= 10n$$

$$S. C. = S. C_{outer} * (S. C_{forward} + S. C_{reverse})$$

$$= (\frac{n+1}{2})(10n + 10n)$$

$$= (\frac{n+1}{2})(20n)$$

$$= (\frac{n+1}{2})(20n)$$

$$= (\frac{20n^2 + 20n}{2})$$

$$= (10n^2 + 10n)$$

Mathematical Analysis

-Big-O efficiency class: O(n^2)

-Definition:

$$10n^{2} + 10n \in O(n^{2})$$
$$10n^{2} + 10n \leq C \cdot n^{2}$$

$$C = 10 + 10 = 20, no = 1$$

 $10n^{2} + 10n \le 20n^{2}$
 $10 + 10 \le 20$
 $20 \le 20 \rightarrow true$

By Definition, we can say that $10n^2 + 10n \in O(n^2)$.

Limit Theorem:

Prove that $10n^2 + 10n \in O(n^2)$ using the limit theorem

$$\begin{split} &\lim_{n \to \infty} \frac{10n^{2} + 10n}{n^{2}} \\ &= \lim_{n \to \infty} \frac{10n^{2}}{n^{2}} + \frac{10n}{n^{2}} \\ &= \lim_{n \to \infty} 10 + \lim_{n \to \infty} \frac{10}{n} \\ &\lim_{n \to \infty} = 10 \ge 0, \end{split}$$

By Limit theorem, we can say that $10n^2 + 10n \in O(n^2)$.

The Alternating Disk Algorithm

Algorithm Design:

Pseudo Code

```
For i = 0 to N+1 step
  if (i % 2 == 0)
     for j = 0 to N-1 step + 2
         if (list[j] > list[j + 1])
            Swap disks
            m++;
         end if
     end for
  else do
     for j = 1 to N-2 step + 2
        if (list[j] > list[j + 1])
            Swap disks
            m++;
        end if
     end for
  end if
end for
```

Step Count

```
For i = 0 to N
                              // (n-0)+1/1= n+1
  if (i % 2 == 0)
                                    // 2 TU
     for j = 0 to N-1 step + 2 // (n-1-0+1)/2 = n/2
        if (list[j] > list[j + 1]) // 1 TU
                                    // 3 TU
           Swap disks
                                    // 1 TU
           M++;
        end
     end
   else
     for j = 1 to N-2 step + 2 // ((n-2)-1+1)/2 = n-2/2
        if (list[j] > list[j + 1]) // 1 TU
           Swap disks
                                    // 3 TU
                                    // 1 TU
           M++;
        end
      end
 End
```

$$S. C. = S. C_{outer} * (S. C_{if block})$$

$$S. C_{outer loop} = n + 1$$

$$S.C_{if\ block} = 2 + max(S.C_{even\ loop}, S.C_{odd\ loop}))$$

S.
$$C_{even loop} = \frac{n}{2} (1 + max(4, 0))$$

= $\frac{n}{2} (1 + 4)$
= $\frac{n}{2} (5)$
= $\frac{5n}{2}$

S.
$$C_{odd \ loop} = \frac{n-2}{2} (1 + max(4, 0))$$

= $\frac{n-2}{2} (1 + 4)$
= $\frac{n-2}{2} (5)$
= $\frac{5n-10}{2}$

S. C =
$$(n + 1)(2 + max(\frac{5n}{2}, \frac{5n-10}{2}))$$

= $(n + 1)(2 + \frac{5n}{2})$
= $\frac{5n^2}{2} + \frac{5n}{2} + 2n + 2$
= $\frac{5n^2}{2} + \frac{9n}{2} + 2$

Mathematical Analysis

-Big-O efficiency class:

-Definition:

$$\frac{5n^{2}}{2} + \frac{9n}{2} + 2 \in O(n^{2})$$

$$\frac{5n^{2}}{2} + \frac{9n}{2} + 2 \leq C \cdot n^{2}$$

$$C = \frac{5}{2} + \frac{9}{2} + 2 = 9, no = 1$$

$$\frac{5n^{2}}{2} + \frac{9n}{2} + 2 \le 9n^{2}$$

$$\frac{5}{2} + \frac{9}{2} + 2 \le 9$$

$$9 < 9 \to true$$

By definition, we can say that $\frac{5n^2}{2} + \frac{9n}{2} + 2 \in O(n^2)$.

Limit Theorem:

Prove that $\frac{5n^2}{2} + \frac{9n}{2} + 2 \in O(n^2)$ using the limit theorem

$$\lim_{n \to \infty} \frac{\frac{5}{2}n^2 + \frac{9}{2}n + 2}{n^2}$$

$$= \lim_{n \to \infty} \frac{\frac{5}{2}n^2}{n^2} + \frac{\frac{9}{2}n}{n^2} + \frac{2}{n^2}$$

$$= \lim_{n \to \infty} \frac{5}{2} + \lim_{n \to \infty} \frac{\frac{9}{2}}{n} + \lim_{n \to \infty} \frac{2}{n^2}$$

$$= \frac{5}{2} + 0 + 0$$

$$= \frac{5}{2}$$

$$\lim_{n \to \infty} = \frac{5}{2} \ge 0,$$

By Limit Theorem, we can say that $\frac{5n^2}{2} + \frac{9n}{2} + 2 \in O(n^2)$.