

Ricardo Gonzalez




rickardo@csu.fullerton.edu

10/9/2022

## Project 1

Github for the project can be found here: <https://github.com/cardogg/project-lawnmover>

### Alternating Disk Problem:

```
Python   Copy  Caption ...  
  
## ALTERNATING DISKS ##  
int numOfSwaps  
disk Sorted = unsorted  
  
for i = 0 in sorted.size() / 2 do  
  for j = i in sorted.size() do  
    if (LIGHT DISK is not followed by DARK DISK)  
      <SWAP LIGHT AND DARK>  
      numOfSwaps++  
  
return sorted and numOfSwaps
```

The pseudocode shows that the worst case time complexity for this would be  $O(n^2)$

In the actual implementation:

```

sorted_disks sort_alternate(const disk_state& before) {
    int numOfSwap = 0; //record # of step swap
    //int state; We can make changes on before
    disk_state after = before;
    //For loop encompasses all light disks -> We can effectively move all of one disk - O(n)
    for (size_t i = 0; i < after.total_count() / 2; i++)
    {
        //For ever light in total until end - O(n)
        for (size_t j = i; j < after.total_count() - 1; j++)
        {
            //swap if color is 1,0 so 110 -> 101
            if (after.get(j) > after.get(j + 1))
            {
                after.swap(j);
                numOfSwap++;
            }
        }
    }

    return sorted_disks(disk_state(after), numOfSwap);
}

```

We can see the overall time is can me shown as  $1 + 1 + (n/2) * (n - 1) * (4 + 1 + 1)$  which we can simplify to be  $6 \frac{n^2}{2} - 6 \frac{n}{2} + 2$

Using limit theorem:

$$\frac{6n^2}{2} - \frac{6n}{2} + 2 \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{6n^2 - 6n + 2}{2n^2} \leftarrow \text{unimportant}$$

$$\frac{d}{dn} \frac{12n - 6}{4n} \rightarrow \frac{d}{dn} \frac{12}{4} = 3$$

$$3 \neq \infty$$

$$\therefore \frac{6n^2}{2} - \frac{6n}{2} + 2 \in O(n^2)$$

Lawnmower Problem:

```

## LAWNMOWER ALGO ##
int numOfSwaps
disk Sorted = unsorted

for i = 0 to sorted.size() - 1 do
    for j in i to sorted.size() - 1 do
        if (LIGHT DISK is not followed by DARK DISK)
            <SWAP LIGHT AND DARK>
            numOfSwaps++

return sorted and numOfSwaps

```

```

sorted_disks sort_lawnmower(const disk_state& before) {
    int numOfSwap = 0;
    disk_state after = before;

    //Loop between the whole vector - O(n)
    for (size_t i = 0; i < after.total_count() - 1; i++)
    {
        //Loops back and fourth between the total of it and perform swaps when needed - O(n)
        for (size_t j = i; j < after.total_count() - 1; j++)
        {
            //Swap like before but remember we are going back and forth
            if (after.get(j) > after.get(j + 1))
            {
                after.swap(j);
                numOfSwap++;
            }
        }
    }

    //Test Comments -- In hindsight the implementation worked find I just wrote before instead of after in the state
    //std::cout << numOfSwap << std::endl;
    //std::cout << after.is_sorted() << std::endl;
    return sorted_disks(disk_state(after), numOfSwap);
}

```

We can see the overall time is can me shown as  $1 + 1 + (n - 1) * (n - 1) * (4 + 1 + 1)$  which we can simplify to be  $6n^2 - 8n + 8$

Using limit theorem:

$$6n^2 - 8n + 8 \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{6n^2 - 8n + 8}{n^2}$$

$$\frac{d}{dn} \frac{6n^2 - 8n + 8}{n^2} \rightarrow \frac{12n - 8}{2n}$$

$$\frac{d}{dn} \frac{12n - 8}{2n} \rightarrow \frac{12}{2} \rightarrow 6 \neq \infty$$

$$\therefore 6n^2 - 8n + 8 \in O(n^2)$$

### Code Compiling and Tests Passing:

```
student@tuffix-vm:~/project-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
```

Tufin 2020 Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Oct 9 03:05

disks.hpp - project-lawnmower - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- PROJECT-LAWNMOWER
- lock.CPSC 335 Project ... U
- lock.CPSC 335 Project 1 Requiem...
- disks\_test
- disks\_test.cpp
- disks.hpp
- LICENSE
- Makefile
- README.md
- rubricTest.hpp

disks.hpp

```
179 }
180 // Algorithm that sorts disks using the lawnmower algorithm.
181 sorted disks sort_lawnmower(const disk_state& before) {
182     int numOfSwaps = 0;
183     disk_state after = before;
184     //Loop between the whole vector - 0(n)
185     for (size_t i = 0; i < after.total_count() - 1; i++)
186     {
187         //Loops back and fourth between the total of it and perform swaps when needed - 0(n)
188         for (size_t j = i; j < after.total_count() - 1; j++)
189         {
190             //Swap like before but remember we are going back and forth
191             if (after.get(j) > after.get(j + 1))
192             {
193                 after.swap(j);
194                 numOfSwaps++;
195             }
196         }
197     }
198     //Test Comments -- In hindsight the implementation worked find I just wrote before instead of after in the state
199     //std::cout << numOfSwaps << std::endl;
200     //std::cout << after.is_sorted() << std::endl;
201     return sorted_disks(disk_state(after), numOfSwaps);
202 }
203
204 disk state:is sorted: passed, score 3/3
205 alternate, n=4: passed, score 1/1
206 alternate, n=3: passed, score 1/1
207 alternate, other values: passed, score 1/1
208 lawnmower, n=4: passed, score 1/1
209 lawnmower, n=3: passed, score 1/1
210 lawnmower, other values: passed, score 1/1
211 TOTAL SCORE = 14 / 14
212
213 student@tuffin-vb:~/project-lawnmower
214 student@tuffin-vb:~/project-lawnmovers
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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@tuffin-vb:~/project-lawnmower  
student@tuffin-vb:~/project-lawnmovers

Ln 170, Col 21 UTF-8 LF C++ Linux

Tufin 2020 Edition [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Oct 9 03:05

disks.hpp - project-lawnmower - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- PROJECT-LAWNMOWER
- lock.CPSC 335 Project ... U
- lock.CPSC 335 Project 1 Requiem...
- disks\_test
- disks\_test.cpp
- disks.hpp
- LICENSE
- Makefile
- README.md
- rubricTest.hpp

disks.hpp

```
158 // Algorithm that sorts disks using the alternate algorithm.
159 sorted disks sort_alternate(const disk_state& before) {
160     //record # of step swap
161     int numOfSwaps = 0;
162     //int state; we can make changes on before
163     disk_state after = before;
164     //For loop encompasses all light disks -> We can effectively move all of one disk - 0(n)
165     for (size_t i = 0; i < after.total_count() / 2; i++)
166     {
167         //For ever light in total until end - 0(n)
168         for (size_t j = i; j < after.total_count() - 1; j++)
169         {
170             //swap if color is 1,0 so 110 -> 101
171             if (after.get(j) > after.get(j + 1))
172             {
173                 after.swap(j);
174                 numOfSwaps++;
175             }
176         }
177     }
178     return sorted_disks(disk_state(after), numOfSwaps);
179 }
180
181 // Algorithm that sorts disks using the lawnmower algorithm.
182 sorted disks sort_lawnmower(const disk_state& before) {
183     int numOfSwaps = 0;
184     disk_state after = before;
185     //Loop between the whole vector - 0(n)
186     for (size_t i = 0; i < after.total_count() - 1; i++)
187     {
188         //Loops back and fourth between the total of it and perform swaps when needed - 0(n)
189         for (size_t j = i; j < after.total_count() - 1; j++)
190         {
191             //Swap like before but remember we are going back and forth
192             if (after.get(j) > after.get(j + 1))
193             {
194                 after.swap(j);
195                 numOfSwaps++;
196             }
197         }
198     }
199     //Test Comments -- In hindsight the implementation worked find I just wrote before instead of after in the state
200     //std::cout << numOfSwaps << std::endl;
201     //std::cout << after.is_sorted() << std::endl;
202     return sorted_disks(disk_state(after), numOfSwaps);
203 }
204
205 disk state:is sorted: passed, score 3/3
206 alternate, n=4: passed, score 1/1
207 alternate, n=3: passed, score 1/1
208 alternate, other values: passed, score 1/1
209 lawnmower, n=4: passed, score 1/1
210 lawnmower, n=3: passed, score 1/1
211 lawnmower, other values: passed, score 1/1
212 TOTAL SCORE = 14 / 14
213
214 student@tuffin-vb:~/project-lawnmower
215 student@tuffin-vb:~/project-lawnmovers
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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@tuffin-vb:~/project-lawnmower  
student@tuffin-vb:~/project-lawnmovers

Ln 170, Col 21 UTF-8 LF C++ Linux