Jessica Vargas
mvrgsj@csu.fullerton.edu
Dylan Nguyen
dylan-nguyen@csu.fullerton.edu
335-04

Team DJ:
Project 1

1. Lawnmower:

Pseudo code:

```
def lawnmower(before):
        disk_state disk = before;

        assert(disk.is_initialized())

        counter=0
        loopAmnt = disk.total_count()/2

        for x = 0 to loopAmnt do
                for i = 0 to loopAmnt do
                        if disk[i] EQUAL DISK_LIGHT AND disk[i+1] EQUAL DISK_DARK      4
                                disk.swap(i)
                                counter = counter + 1
                        endif
                endfor
                for j = disk.total_count() - 1 to 0 do
                        if disk[j] EQUAL DISK_DARK AND disk[j-1] EQUAL DISK_LIGHT      4
                                disk.swap(j-1)
                                counter = counter + 1
                        endif
                endfor
        endfor
return disk, counter
```

$$\sum_{x=0}^{n/2}\left(1+\sum_{i=0}^{n/2}4+\sum_{j=n-1}^{0}4\right)$$

$$\sum_{x=0}^{n/2}\left(1+4\left(\tfrac{n}{2}-0+1\right)+4\left(0-(n-1)+1\right)\right)$$

$$\left(1+2n-0+4+8-4n\right)$$

$$\sum_{x=0}^{n/2} -2n+13$$

$$\sum_{x=0}^{n/2}-2n+\sum_{x=0}^{n/2-1}13$$

$$-2n\left(\tfrac{n}{2}-0+1\right)+13\left(\left(\tfrac{n}{2}-1\right)-0+1\right)$$

$$-n^2-2n+\tfrac{13}{2}n+5$$

$$-n^2 \rightarrow O(n^2)$$

Mathematical Analysis:

For the for loop from x = 0 to n/2 there are two nested for loops. The first for loop goes from i = 0 to n/2 and the second one starts as j= n-1 to 0. The nested for loops both share the same amount of inner steps (4) and are treated as such. After adding an additional 1 for initial step count we solve up to the outer for loop with -2n+13 steps on the inside. We distribute the sigma to both sides to solve and we end with -n^2 -2n +13/2 n. We add 5 to the end to represent the steps outside the loops.

We drop the dominated terms and are left with -n^2. Lastly we drop the multiplicative constant of -1 and are left with n^2, giving us O(n^2). Therefore the lawnmower algorithm is initialized with an efficiency of O(n^2).

2. Alternating:

Pseudo code:

```
def sort_alternate(disk):

    count=0
    pos=0
    n=disk.total_count()
    assert.(disk.is_initialized())

    for i to n do

      for f=0 to disk.dark_count() do
          if disk[pos] == disk_light AND disk[pos+1] == disk_dark
              disk.swap(pos)
              count = count + 1                                    5
          endif
        pos = pos + 2
      endfor
      pos = 1

      for g=0 to disk.dark_count()-1 do
          if disk[pos] == disk_light AND disk[pos+1] == disk_dark
              disk.swap(pos)                                       5
              count = count + 1
          endif
        pos = pos + 2
      endfor
      pos = 0

    endfor

return disk, count
```

$$\sum_{i=0}^{n}\left(3+\sum_{f=0}^{n}5+\sum_{g=0}^{n-1}5\right)$$

$$\sum_{i=0}^{n}\left(3+5(n-0+1)+5(n-1-0+1)\right)$$

$$\sum_{i=0}^{n}\left(3+(5n+5)+5n\right)$$

$$\sum(10n+8) \rightarrow \sum^{n} 10n + \sum^{n-1} 8$$

$$10n(n-0+1)+8(n-1-0+1)$$

$$10n^2+10n+8n \rightarrow 10n^2+18n$$

$$10n^2+18n+5 \quad \text{(5 from outer)}$$

$$10n^2+18n$$

$$10n^2 \rightarrow O(n^2)$$

Mathematical Analysis:

For loop goes from 0 to n with two inner loops. First inner loop goes f=0 to n. The second loop starts g=0 to n-1. Both loops have inner step count and overhead totaling to 5. Add 3 for the outer loop step counts. Evaluate the sigmas using C(L-F+1) and add 3 to get 10n+8. Distribute the sigma to bother sides to get and add 5 for the outer step count to get 10n^2 + 18n + 5.

Drop the dominant term 5 and 18n to get 10n^2. Then drop the multiplicative constant 10 according to the properties of O(n) to get n. Therefore, the efficiency of sort_alternate() is O(n^2).

3. is_initialized()

Pseudo code:

```
def is_initialized():

        disk_color colorCheck = DISK_LIGHT
        n= _colors.size()

        for i=0 to n do
                if _colors[i] NOT EQUAL colorCheck
                    return false
                endif
                if colorCheck EQUAL DISK_LIGHT
                    colorCheck = DISK_DARK
                endif
                else
                    colorCheck=DISK_LIGHT
        endfor
return true
```

6

$$3 + \sum_{i=0}^{n} 6 \qquad C(L-F+1)$$

$$6(n-0+1)$$

$$6n+6$$

$$6n+6+3 = 6n+9$$

$$6n \rightarrow O(n)$$

Mathematical Analysis:

For loop starts i=0 to n. Evaluate the sigma to C(L-F+1) and add the outer step count 3 to get 6n+9. Drop the dominant term 9 to get 6n. Then drop the multiplicative constant 6 according to the properties of O(n) to get n. Therefore, the efficiency of is_initialized() is O(n).

4. is_sorted()

Pseudo code:

```
def is_sorted():
        colorsHalf = total_count()/2  1

        for j=0 to colorsHalf do  1
                if _colors[j] EQUAL DISK_LIGHT  1  ] 3
                        return false  1
                endif
        endfor

        for i = colorsHalf to total_count() do  1
                if _colors[i] EQUAL DISK_DARK  1  ] 3
                        return false  1
                endif
        endfor
return true  1
```

$$\sum_{j=0}^{n/2} 3 + \sum_{i=n/2}^{0} 3 + 2$$

$$3\left(\frac{n}{2} - 0 + 1\right)$$

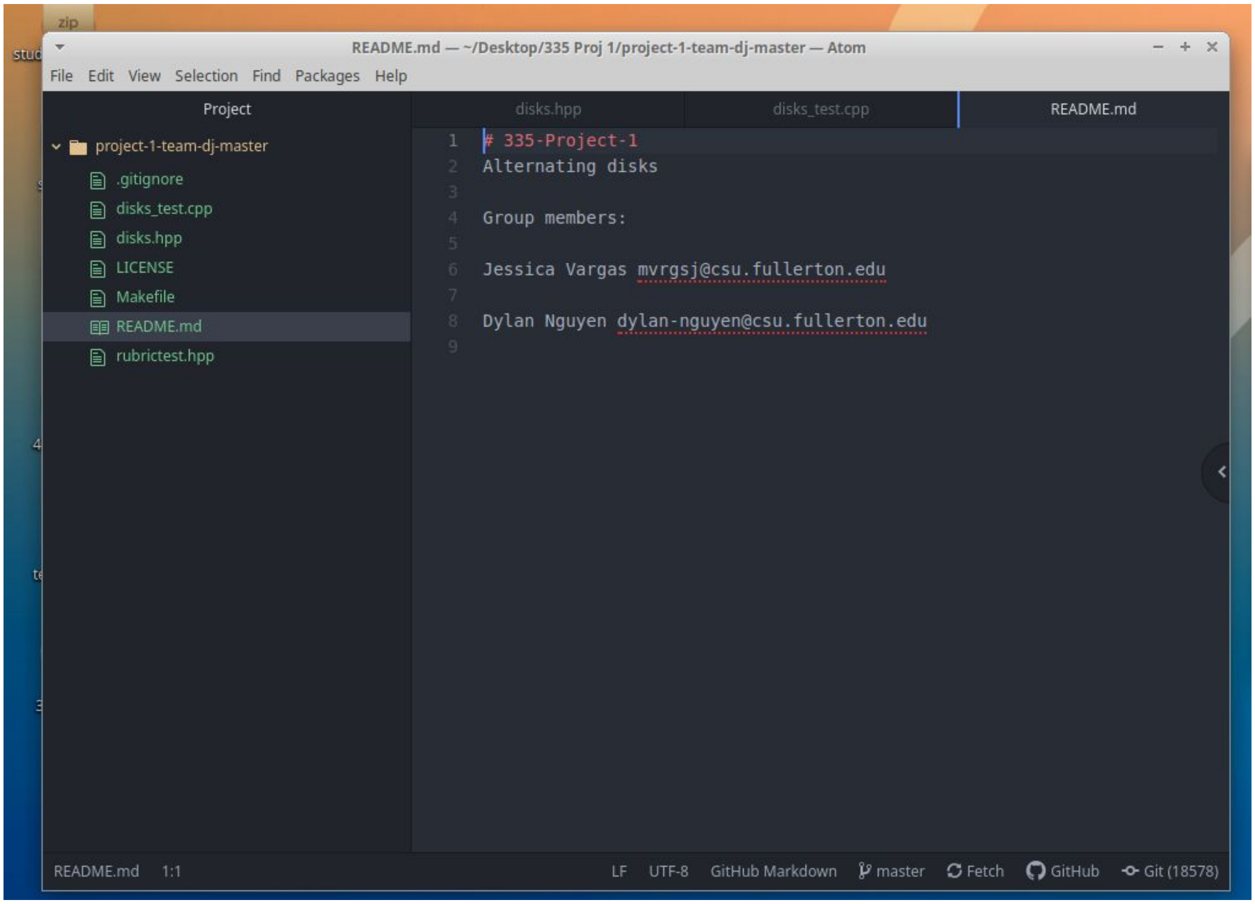
$$\left(\frac{3n}{2} + 3\right)$$

$$O(n)$$

Mathematical Analysis:

For the algorithm is_sorted, we have two for loops of equal length and steps. To calculate big O notation easier we only need to choose one of the loops to calculate. As such we chose the first loop. We add one initial count to the body count and multiply it by 3. After that we are left with 3n/2 +3, we drop dominated terms and the result is 3n/2. When we drop the multiplicative constant we are left with simply n, resulting in O(n).

<u>Screenshots</u>:




```
student@tuffix-vm:~/Desktop/335 Proj 1/project-1-team-dj-master$ 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 1/1
disk_state::is_sorted: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, n=4: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 10 / 10
```