Project 1 Report

Alex Lu
Email: alexluc2001@csu.fullerton.edu

Stefan Parrish
Email: parrishstefan@csu.fullerton.edu

README.md × | disks_test.cpp | disks.hpp M | main.cpp

Proj1 > alternating-disks-stefan-parrish > README.md > # proj-1-alternating-disks

```
1  # proj-1-alternating-disks
2  Project 1 alternating disks DLDL LLDD
3
4  Group members:
5
6  Stefan Parrish parrishstefan@csu.fullerton.edu
7
8  Alex Lu alexluc2001@csu.fullerton.edu
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
shico@DESKTOP-U03ERS1:/mnt/c/Users/Stefan/Documents/School/CS335/Proj1/alternating-disks-stefan-parrish$ 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
lawnmower, n=4: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, other values: passed, score 1/1
alternate, n=4: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, other values: passed, score 1/1
TOTAL SCORE = 14 / 14
```

Pseudocode:

```
sort_lawnmower(disk_state before) {
    Set disk_state parameter to after
    Set integer swapCount to 0
    Set integer totalCount to after.totalcount()
    Set integer rotation to 0
    Set integer loops to 0

while(The disks aren't sorted)
For i=rotation to i < ((totalCount-1)-rotation) do
If(after.get(i) == DISK_DARK and after.get(i+1) == DISK_LIGHT)
           after.swap(i)
           swapCount += 1
      Endif
      i+=1
Endfor
rotation+=1
For j=((totalCount-1)-rotation) to j > rotation do
if(after.get(j) == DISK_LIGHT and after.get(j-1) == DISK_DARK)
           after.swap(j-1)
           swapCount+=1
      Endif
Endfor
rotation+=1
loops+=1
      Endwhile

      return sorted_disks(after, swapCount)

sort_alternate(disk_state before)
      Set disk_state parameter to after
       Set integer swapCount to 0
       Set integer totalCount to after.totalcount()
for(Iterate through the disks)

      For i=0 to i<totalCount do
           For j=i to j<(totalCount-1) do
if(after.get(j) == DISK_DARK && after.get(j + 1) == DISK_LIGHT)
      after.swap(j)
       swapCount+=1
Endif
j+=1
           Endfor
           totalCount-=1
```

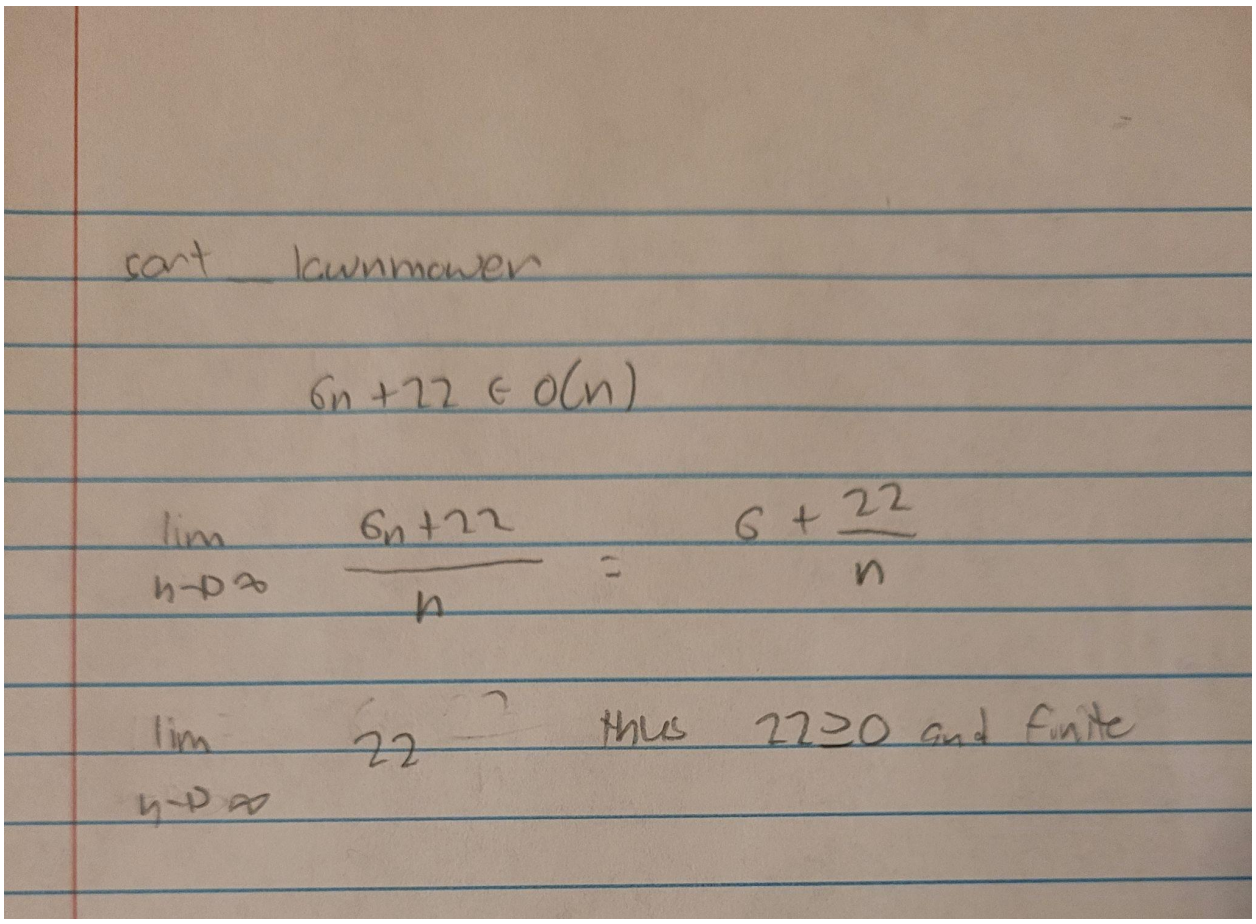
```
      i+=1
Endfor

return sorted_disks(after, swapCount)
```

Time Complexity:

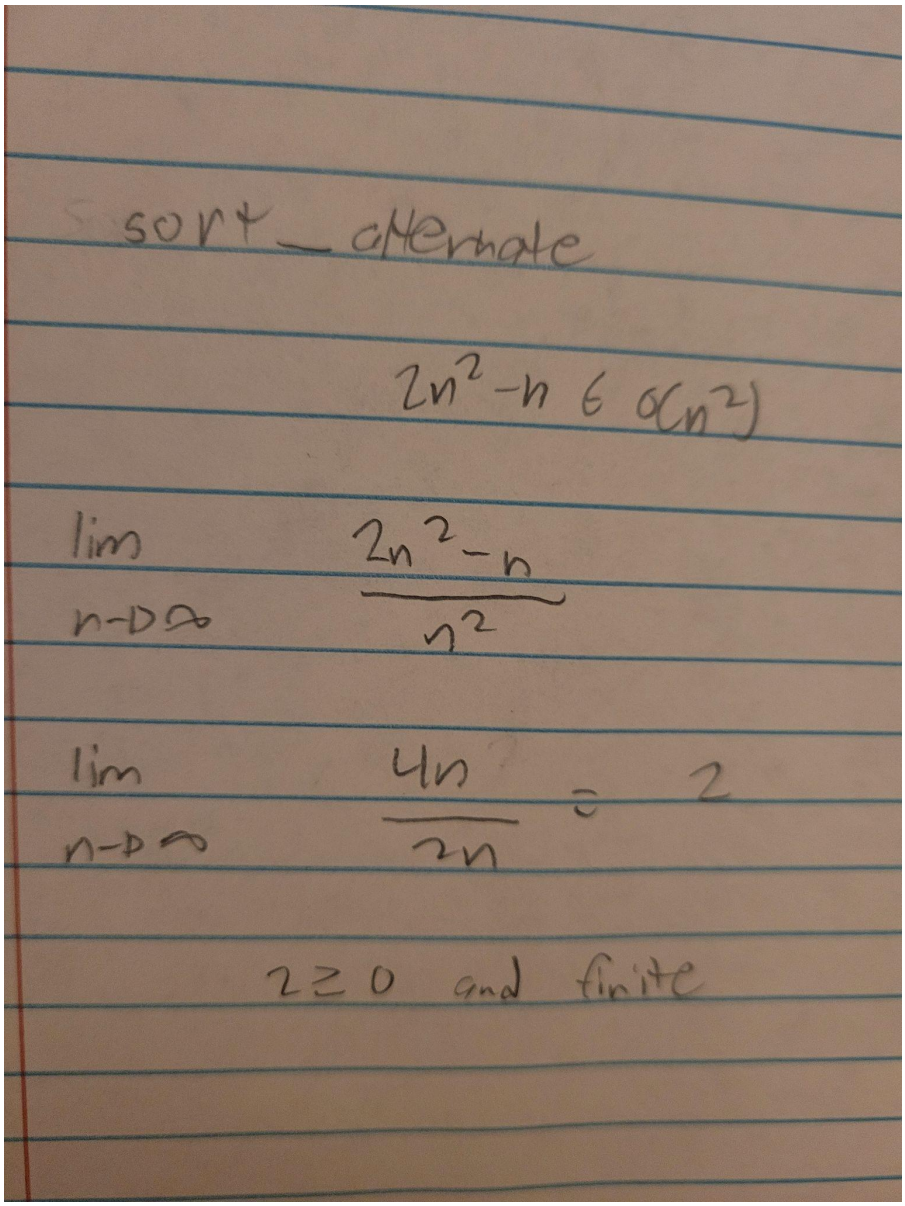
sort_lawnmower() = O(n).

Given sort_lawnmower()'s Step Count = 6n+22
This step count is calculated within the code's comments and with that, we can prove the time complexity of sort_lawnmower(). Since the step count contains "6n" we can assume that it may be O(n). Using the limit theorem, we calculated this:



meaning, 6n+22∈O(n).

Given sort_alternate()'s Step Count = $2n^2-n$
This step count is calculated within the code's comments and with that, we can prove the time complexity of sort_alternate(). Since the step count contains "$2n^2$" we can assume that it may be $O(n^2)$. Using the limit theorem, we calculated this:



meaning, $2n^2-n \in O(n^2)$.