

In this Project we were given an alternating disk problem to which we had to use two different algorithms to arange the disk colors in the right location. Light disks on the left and right disks on the right. We wrote psudocode which we turned into real code which let us compile and pass the tests. We later made a mathematical analysis to proof our time efficiency.

### Screenshot Atom editor

The screenshot shows the Atom code editor interface. The title bar reads "disks.hpp — ~/Downloads — Atom". The menu bar includes File, Edit, View, Selection, FInd, Packages, and Help. On the left, there's a sidebar titled "Project" showing a file tree with "Downloads", "Project\_1\_335-main" (containing "disks\_test.cpp", "disks.hpp", "README.md", "rubrictest.hpp", "run", "code\_1.72.0-1664926972\_amd64.deb", "Project\_1\_335-main.zip", and "project-lawnmover-main.zip"), and "GitHub" and "Git (0)". The main editor area has two tabs: "disks.hpp" and "README.md". The "disks.hpp" tab contains C++ code for sorting disks. The "README.md" tab contains project metadata. The status bar at the bottom shows "Project\_1\_335-main/disks.hpp 171:1" and icons for LF, UTF-8, C++, GitHub, and Git (0).

```
disks.hpp
156     int count = 0;
157     for(size_t i = 0; i < after.dark_count() + 1; i++)
158     {
159         for(size_t j = i; j < after.total_count() - 1; j++)
160         {
161             if (after.get(j) == DISK_DARK && after.get(j + 1) == DISK_LIGHT)
162             {
163                 after.swap(j);
164                 count++;
165             }
166         }
167     }
168
169     return sorted_disks(after, count);
170 }
171
172 // Algorithm that sorts disks using the lawnmower algorithm
173 sorted_disks sort_lawnmower(const disk_state& before)
174 {
175     int count = 0;
176     disk_state after = before;
177
178     for(size_t i = 0; i < after.total_count() / 2; i++)
179     {
180         for(size_t j = 0; j < after.total_count() - 1; j++)
181         {
182             if(after.get(j) == DISK_DARK && after.get(j + 1) == DISK_LIGHT)
183             {
184                 after.swap(j);
185             }
186         }
187     }
188
189     return sorted_disks(after, count);
190 }
```

```
README.md
1 # Project 1_335
2 CPSC 335 Project 1
3
4 anahid zandi haghghi | anazandi@csu.fullerton.edu
5 Daniel Palacio | Palacio796@csu.fullerton.edu
6
7
8
```

## Screenshot Code Running

The screenshot shows a Linux desktop environment with a terminal window open in the Atom code editor. The terminal window displays the output of a C++ program named `disks_test.cpp`. The program runs a series of tests on a class `disk_state` and prints a total score. The terminal window is titled "disks.hpp" and is part of a project named "Project\_1". The Atom interface includes a sidebar with icons for various applications like a file manager, terminal, and browser.

```
daniel@daniel-Latitude-3301:~/Downloads/Project_1_335-main$ clang++ disks_test.cpp
daniel@daniel-Latitude-3301:~/Downloads/Project_1_335-main$ ./run
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
ruf'alternate, n=3: passed, score 1/1
run 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

daniel@daniel-Latitude-3301:~/Downloads/Project_1_335-main$
```

The code editor shows the source code for `disks.hpp`, specifically lines 189 to 195, which implement a swap operation on a `disk_state` object.

## Alternating Algorithm Pseudocode:

```
Count = 0
For i = 0 to (n/2) +1 do
    For j = i to n-1 step 2
        if Disk [j] == Dark && Disk[j+1] == Light
            swap(Disk[j], Disk[j+1])
            Count++
    End if
End for
End for
```

## Mathematical analysis

Alternating algorithm (math analysis)

count = 0  $\rightarrow$  1 tu

for  $i=0$  to  $(\frac{n}{2})+1$  do

    for  $j=i$  to  $n-1$  step 2

        if  $disk[j] == \text{Dark}$  &  $disk[j+1] == \text{Light}$   $\rightarrow$  4 tu

        Swap( $disk[j]$ ,  $disk[j+1]$ )  $\rightarrow$  5 tu  $\rightarrow$  taken from Swap function

        Count ++  $\rightarrow$  1 tu

    end if

end for

end for

$$\sum_{i=0}^{\left(\frac{n}{2}\right)} \sum_{j=i}^{\left(\frac{n-1}{2}\right)} q \rightarrow \sum_{j=i}^{\left(\frac{n-1}{2}\right)} q \rightarrow \sum_{j=1}^{\left(\frac{n-1}{2}\right)} q - \sum_{j=1}^{i-1} q = \frac{q(n-1)}{2} - q(i-1) \\ = \frac{qn - 18i + q}{2}$$

$$\sum_{i=0}^{\frac{n}{2}} \frac{qn}{2} - \sum_{i=0}^{\frac{n}{2}} \frac{18i}{2} + \sum_{i=0}^{\frac{n}{2}} \frac{q}{2} = \left(\frac{n}{2}+1\right) \times \left(\frac{qn}{2}\right) = \frac{qn(2+n)}{4}$$

constant series

$$\sum_{i=0}^{\frac{n}{2}} q_i = q \left( \frac{\left(\frac{n}{2}+1\right)\left(\frac{n}{2}+1+1\right)}{2} \right) = \frac{q(2+n)(4+n)}{8}$$

$$\sum_{i=0}^{\frac{n}{2}} \frac{q}{2} = \left(\frac{n}{2}+1\right) \left(\frac{q}{2}\right) = \frac{q(2+n)}{4} = \left(\frac{q_n(2+n)}{4}\right) - \left(\frac{q(2+n)(4+n)}{8}\right) + \left(\frac{q(2+n)}{4}\right) \\ \left(\frac{q_n^2 + 27n + 18}{4}\right) - \left(\frac{q(2+n)(4+n)}{8}\right) = \frac{q_n^2 - 36 + 1}{8} \stackrel{\text{count=0}}{=} \left(\frac{q_n^2 - 28}{8} + 1\right)$$

$$f(n) = \frac{9n^2 - 28}{2}$$

$$\frac{9n^2 - 28}{2} \in O(n)$$

$$\lim_{n \rightarrow \infty} \frac{9n^2 - 28}{2n} \stackrel{\text{1. Schiffler}}{\rightarrow} \lim_{n \rightarrow \infty} \frac{18n}{2} = \infty = f(n) \notin O(n)$$

$$\frac{9n^2 - 28}{2} \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{9n^2 - 28}{2n^2} = \lim_{n \rightarrow \infty} \frac{18n}{4n} = \lim_{n \rightarrow \infty} \frac{18}{4} - \frac{28}{2n^2} = f(n) \in O(n^2)$$

We can conclude that the Alternate Algorithm efficiency class is  $O(n^2)$

### Lawnmower Algorithm PseudoCode:

```

Int count = 0
For i = 0 to n/2 do
    For j = 0 to n-1 do
        If Disk[j] == dark && Disk[j] != Disk[j+1]
            swap ( Disk[j] , Disk[j+1] )
            count ++
        End if
    End for
    For j = n - 1 to 0 step -1
        If Disk[j] == dark && Disk[j] != Disk[j-1]
            swap ( Disk[j-1], Disk[j] )
            Count ++
        End it
    End for
End for

```

### Mathematical Analysis:

S-C-Sum (for (block) \* for (block))

```

Count = 0 → 1 TU
for i=0 to  $\frac{n}{2}$  do →  $(\frac{n}{2}) - 0 + 1 = (\frac{n}{2} + 1)$  times
  for j=0 to n-1 do →  $(n-1) - 0 + 1 = n$  times
    if disk[j] == Dark & disk[j] != disk[j+1] → 4 TU
      Swap (disk[j], Disk[j+1]) → 5 TU
      Count ++ → 1 TU
    end if
  end for
  for s=n-1 to 0 step -1 →  $n - 0 + 1 = n$  times
    if disk[s] == Light & disk[s] != disk[s-1] → 4 TU
      Swap (disk[s-1], disk[s]) → 5 TU
      Count ++ → 1 TU
    end if
  end for
end for

```

$$S-C = \left( \frac{n}{2} + 1 \right) (n \cdot (4 + \max(s, 1))) (n - (4 + \max(s, 1)))$$

$$= \left( \frac{n}{2} + 1 \right) (9n) (9n) = 81n^2 \left( \frac{n}{2} + 1 \right) = \frac{81n^2(2+n)}{2}$$

$$f(n) = \frac{81n^2(2+n)}{2}$$

$$\frac{81n^2(2+n)}{2} \in O(n)$$

$$\lim_{n \rightarrow \infty} \frac{81n^2(2+n)}{2n} \Rightarrow \lim_{n \rightarrow \infty} \frac{81n^2}{2n} + \lim_{n \rightarrow \infty} \frac{(2+n)}{2n} \therefore \frac{81n^2(2+n)}{2} \notin O(n)$$

$$\frac{81n^2(2+n)}{2} \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{81n^2(2+n)}{2n^2} \Rightarrow \lim_{n \rightarrow \infty} \frac{81n^2}{2n^2} + \lim_{n \rightarrow \infty} \frac{(2+n)}{2n^2}$$

$$\therefore \frac{81n^2(2+n)}{2} \in O(n^2)$$

We can conclude that the lawnmower Algorithm efficiency class is  $O(n^2)$