

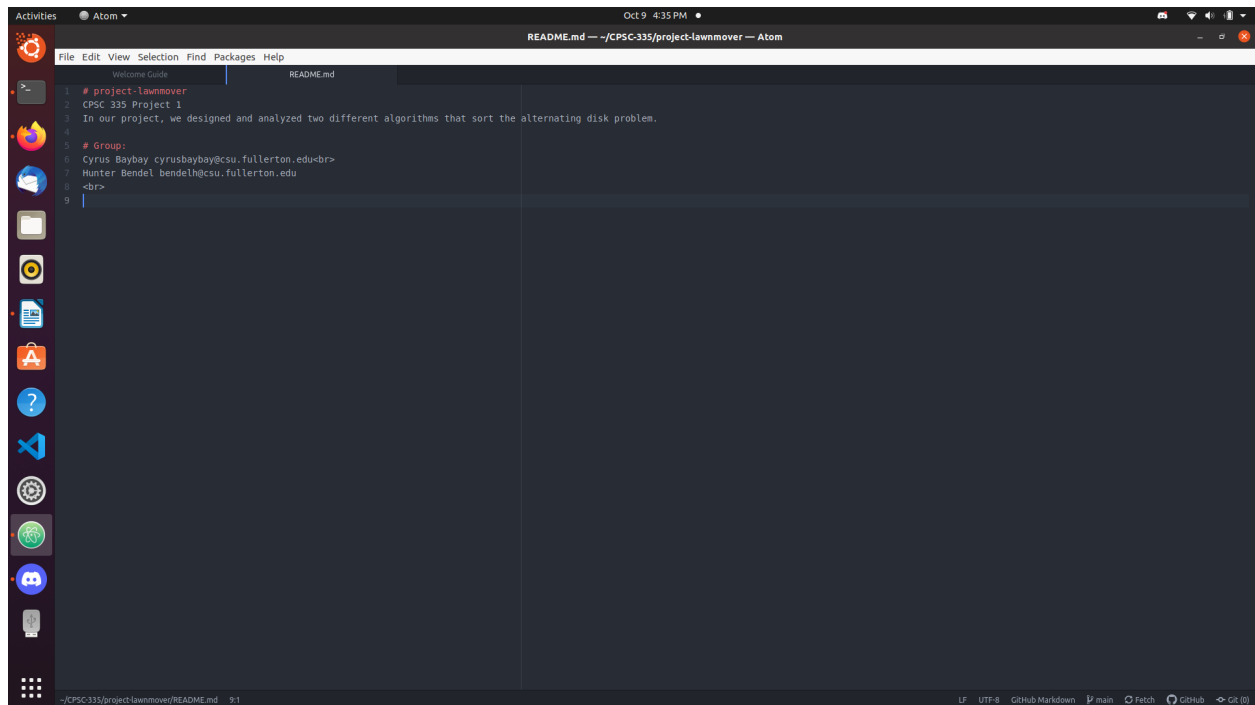
CPSC 335 Project 1 Design & Analysis

Emails

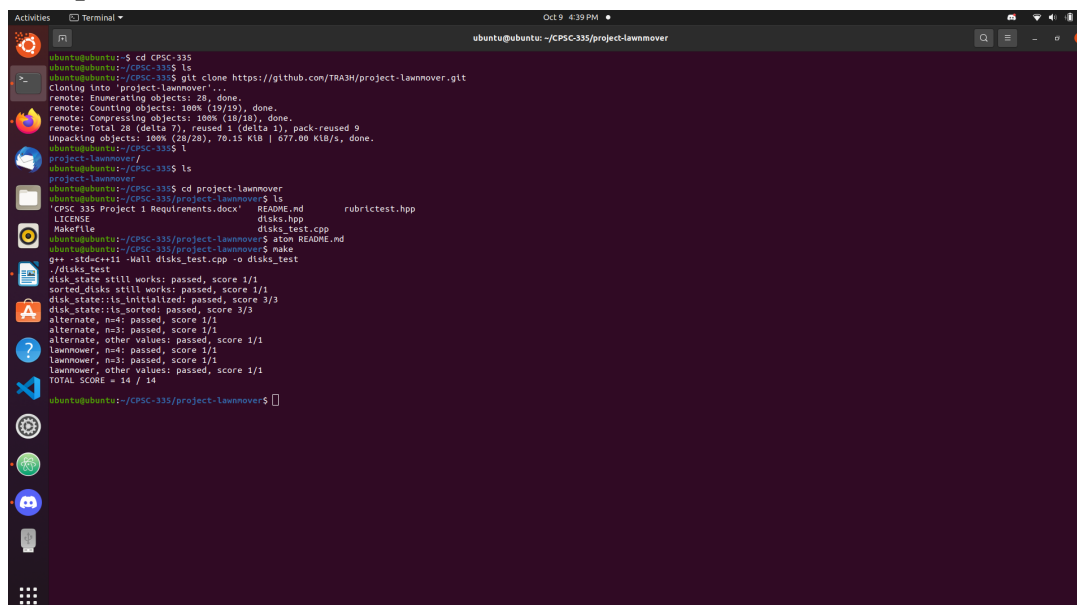
cyrusbaybay@csu.fullerton.edu

bendelh@csu.fullerton.edu

Tuffix Screenshot



Compile Screenshot



Lawnmower Algorithm

Pseudocode:

```

14  ## Lawnmower Algorithm
15
16  int numOfSwap
17  disk_state after = before
18
19  for i = 0 to after.size() - 1 do
20      for j in i to after.size() - 1 do
21          if Dark Disk does not come after Light Disk do
22              swapOperation
23              numOfSwap++
24
25  return after and numOfSwap

```

Implementation:

```

174  // Algorithm that sorts disks using the lawnmower algorithm.
175  sorted_disks sort_lawnmower(const disk_state& before) {
176      int numOfSwap = 0;
177      disk_state after = before;
178
179      // Loop over the entire list at O(n) complexity
180      for (size_t i = 0; i < after.total_count() - 1; i++) {
181          // Loop left-to-right and right-to-left over the list at O(n) complexity
182          for (size_t j = 1; j < after.total_count() - 1; j++) {
183              // Swap if it goes dark(1), then light(0), but account for going both directions
184              if (after.get(j) > after.get(j + 1)) {
185                  after.swap(j);
186                  numOfSwap++;
187              }
188          }
189      }
190
191      return sorted_disks(disk_state(after), numOfSwap);
192  }

```

Mathematical Analysis:

Lawnmower Algorithm

$$\begin{aligned}
 &1 + 1 + (n-1) \cdot (n-1) \cdot (4+1+1) = \\
 &2 + (n-1) \cdot (n-1) \cdot (6) = \\
 &2 + 6(n-1)^2 = \\
 &= 6n^2 - 12n + 8
 \end{aligned}$$

Annotations: "for outer" points to the first $(n-1)$, "for inner" points to the second $(n-1)$, and "if" points to the $(4+1+1)$.

Using limit theorem:

$$6n^2 - 12n + 8 \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{6n^2 - 12n + 8}{n^2}$$

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

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

Therefore: $6n^2 - 12n + 8 \in O(n^2)$ ✓

Alternate Algorithm

Pseudocode:

```

1  ## Alternate Algorithm ##
2
3  int numOfSwap
4  disk_state after = before
5
6  for i = 0 in after.size() / 2 do
7      for i = j in after.size() do
8          if Dark Disk does not come after Light Disk do
9              swapOperation
10             numOfSwap++
11
12  return after and numOfSwap

```

Implementation:

```

153 // Algorithm that sorts disks using the alternate algorithm.
154 sorted_disks sort_alternate(const disk_state& before) {
155     int numOfSwap = 0; //record # of step swap
156     // Initializing another disk state will allow us to work on a list identical to before
157     disk_state after = before;
158     // The below loop will cover all light disks at O(n) complexity
159     for (size_t i = 0; i < after.total_count() / 2; i++) {
160         // Iterates for every light in the list at O(n) complexity
161         for (size_t j = i; j < after.total_count() - 1; j++) {
162             // Swap if it goes dark(1), then light(0)
163             if (after.get(j) > after.get(j + 1)) {
164                 after.swap(j);
165                 numOfSwap++;
166             }
167         }
168     }
169
170     return sorted_disks(disk_state(after), numOfSwap);
171 }

```

Mathematical Analysis:

Alternate Algorithm:

$$\begin{aligned}
 &1+1+(n/2) \cdot (n-1) \cdot (1+1+4) \\
 &= 2 + (n/2) \cdot (n-1) \cdot (6) \\
 &= 2 + 6 \left(\frac{n^2}{2} - \frac{n}{2} \right) \\
 &= 6 \frac{n^2}{2} - 6 \frac{n}{2} + 2
 \end{aligned}$$

Using limit theorem to prove:

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

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

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

Therefore:

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