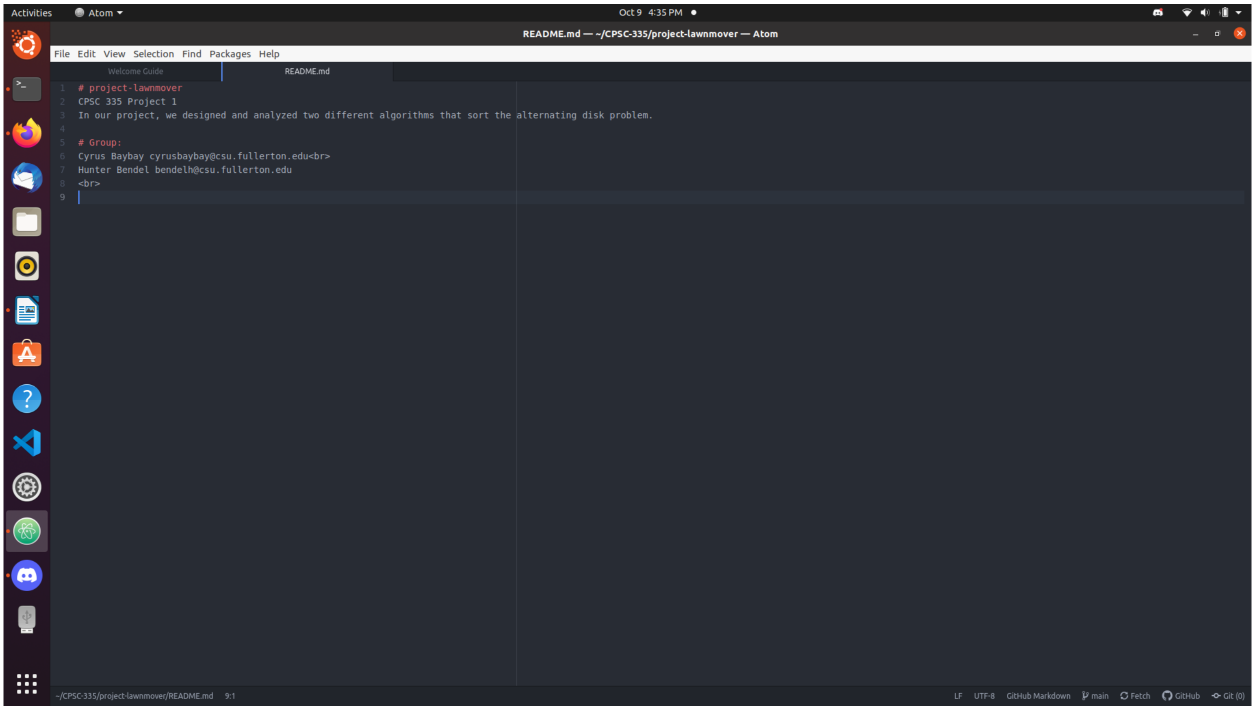
CPSC 335 Project 1 Design & Analysis

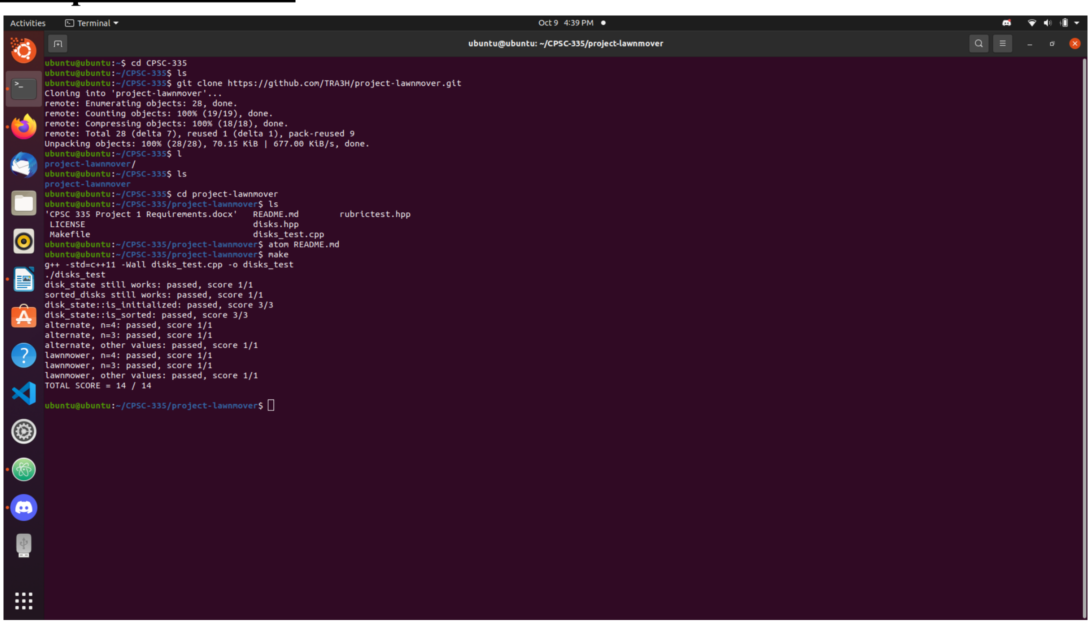
Emails

cyrusbaybay@csu.fullerton.edu
bendelh@csu.fullerton.edu

Tuffix Screenshot



Compile Screenshot



Lawnmower Algorithm

Pseudocode:

```
## Lawnmower Algorithm

int numOfSwap
disk_state after = before

for i = 0 to after.size() - 1 do
    for j in i to after.size() - 1 do
        if Dark Disk does not come after Light Disk do
            swapOperation
            numOfSwap++

return after and numOfSwap
```

Implementation:

```
// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state& before) {
  int numOfSwap = 0;
  disk_state after = before;

  // Loop over the entire list at O(n) complexity
  for (size_t i = 0; i < after.total_count() - 1; i++) {
    // Loop left-to-right and right-to-left over the list at O(n) complexity
    for (size_t j = 1; j < after.total_count() - 1; j++) {
      // Swap if it goes dark(1), then light(0), but account for going both directions
      if (after.get(j) > after.get(j + 1)) {
        after.swap(j);
        numOfSwap++;
      }
    }
  }

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

Mathematical Analysis:

Lawnmower Algorithm

$$1 + 1 + \overset{\text{for outer}}{(n-1)} \cdot \overset{\text{for inner}}{(n-1)} \cdot \overset{\text{if}}{(4+1+1)} =$$

$$2 + (n-1) \cdot (n-1) \cdot (6) =$$

$$2 + 6(n-1)^2 =$$

$$= 6n^2 - 12n + 8$$

Using limit theorem:

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

$$\int_{n \to \infty} \frac{6n^2 - 12n + 8}{n^2}$$

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

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

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

Alternate Algorithm

Pseudocode:

```
## Alternate Algorithm ##

int numOfSwap
disk_state after = before

for i = 0 in after.size() / 2 do
    for i = j in after.size() do
        if Dark Disk does not come after Light Disk do
            swapOperation
            numOfSwap++

return after and numOfSwap
```

Implementation:

```
// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {
  int numOfSwap = 0;                                                    //record # of step swap
  // Initializing another disk state will allow us to work an a list identical to before
  disk_state after = before;
  // The below loop will cover all light disks at O(n) complexity
  for (size_t i = 0; i < after.total_count() / 2; i++) {
    // Iterates for every light in the list at O(n) complexity
    for (size_t j = i; j < after.total_count() - 1; j++) {
      // Swap if it goes dark(1), then light(0)
      if (after.get(j) > after.get(j + 1)) {
        after.swap(j);
        numOfSwap++;
      }
    }
  }

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

Mathematical Analysis:

Alternate Algorithm:

$$1+1+(n/2)\cdot(n-1)\cdot(1+1+4)$$

$$= 2+(n/2)\cdot(n-1)(6)$$

$$= 2+6\left(\frac{n^2}{2}-n/2\right)$$

$$= 6\frac{n^2}{2}-6\frac{n}{2}+2$$

Using limit theorem to prove:

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

$$\int_{n\to\infty} \frac{6n^2-6n+2}{2n^2}$$

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

Therefore:

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