CPSC 335 - Nelson Luong, nluong@csu.fullerton.edu
Project 1 Submission

p-code for left_to_right algorithm

```
n = total number of whites
sort_left_to_right (V):
        initialize variables                                        (1 step)
        for i from 0 to n                                           (n steps)
                x = i # reset position back to the left after swaps (1 step)
                for j = x from n - x while j increments by 2        (n step)
                        V.swap left index with index+1 (right)      (1 step)
                        Increment swap count                        (1 step)
        Return                                                      (1 step)
```

Code for Algorithm

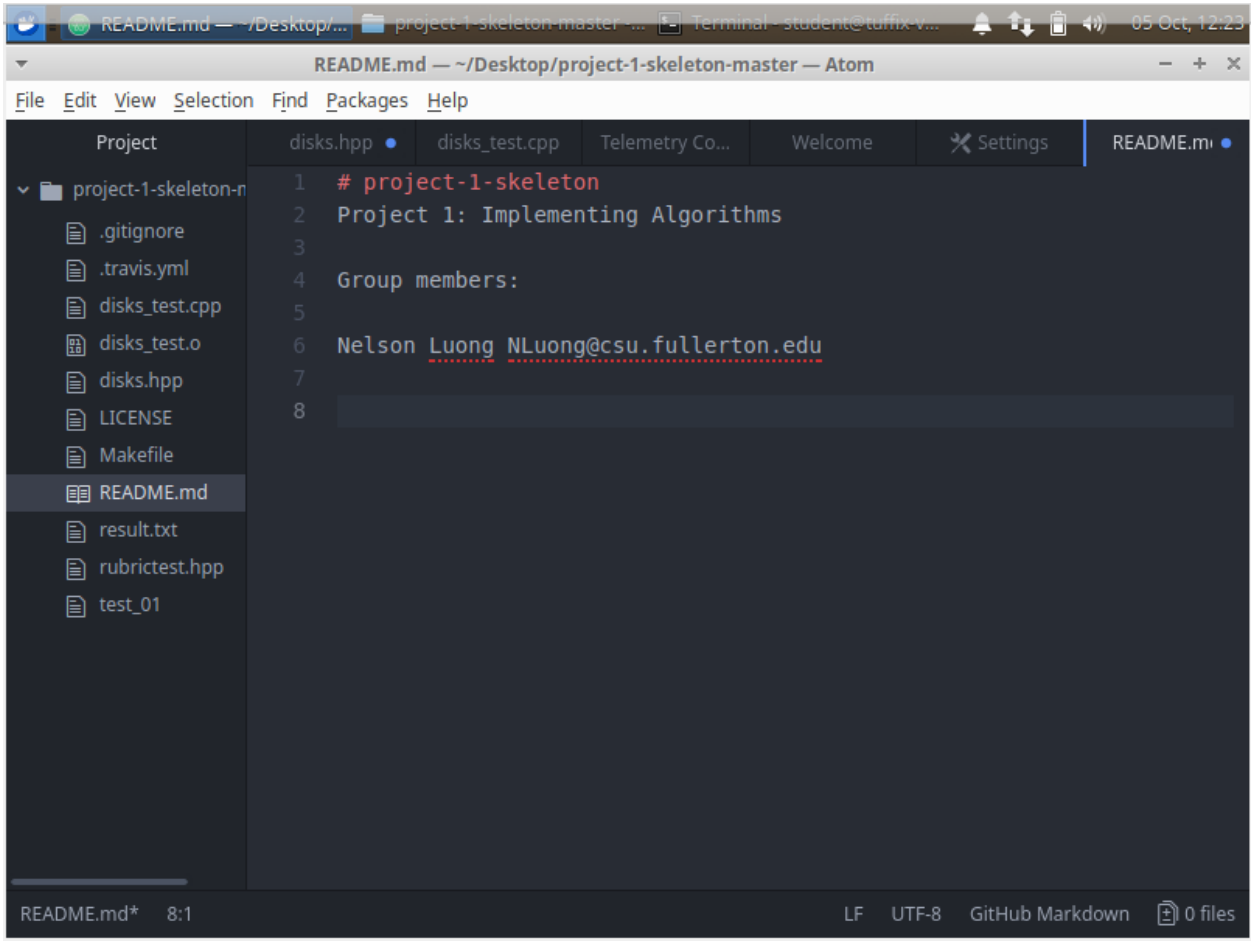
```
sorted_disks sort_left_to_right(const disk_state& before) {
        int x = 0;
        int swaps = 0;
        for (int i = 0; i < (before.total_count()/2); i++) {
                x = i; // reset back to the left
                for (int j = x; j < (before.total_count() - x); j += 2) {
                        before.swap(j);
                        swaps++;
                }
        }

  return sorted_disks(before, swaps);
```

P-code for Lawnmower method

```
Sort_lawnmower(V)
# incomplete
```

CPSC 335 - Nelson Luong, nluong@csu.fullerton.edu
Project 1 Submission

Screenshot



Time Complexity
Analysis for Left to Right
T(n) = 1 + (n(1(n + 1 + 1))) + 1
= n(n+1+1) +1
= n^2 + n + n + 1
= n^2 + 2n + 1
= n^2 + n
= O(n^2 + n)
= O(n^2)

Analysis for Lawnmower
#Incomplete