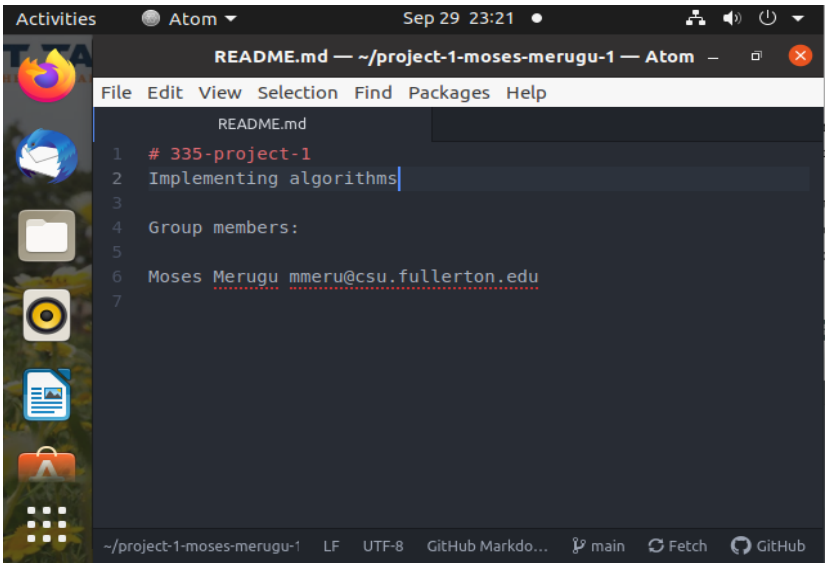
Moses Merugu

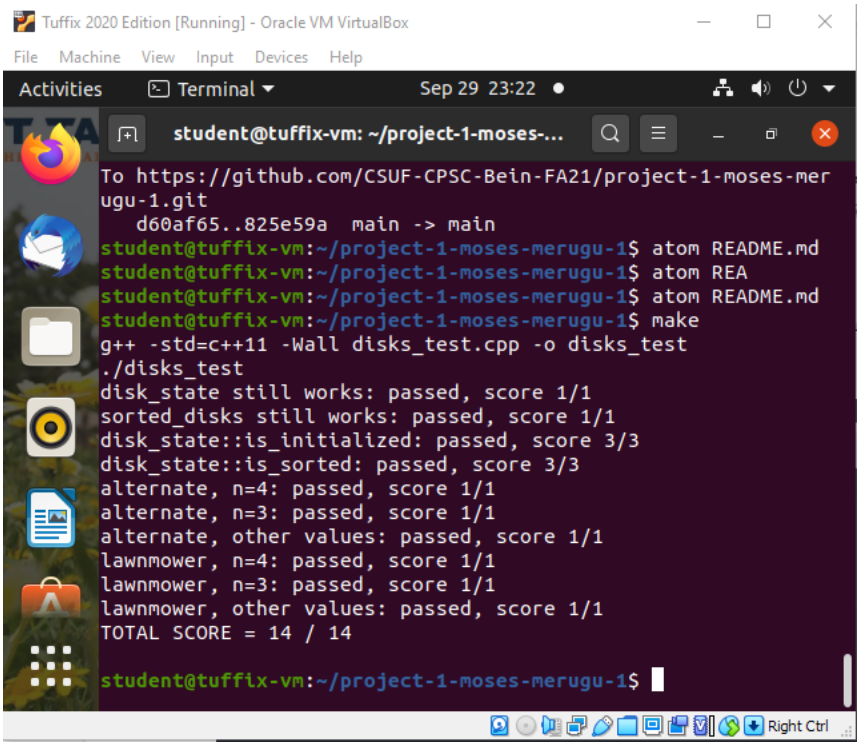
mmeru@csu.fullerton.edu

10/1/2021

CPSC 353-01

Project 1 Report





Lawnmower

Pseudocode

Count = 0

Disk_state = before

from i to light_count

if (i % 2 == 0) go left to right

if adjacent disks differ disk_state.get(j) > disk_state.get(j+1)

swap() and increment swap count

else go right to left

if adjacent disks differ disk_state.get(j) > disk_state.get(j+1)

swap() disk state and increment swap count

Return sorted disks state and swap count

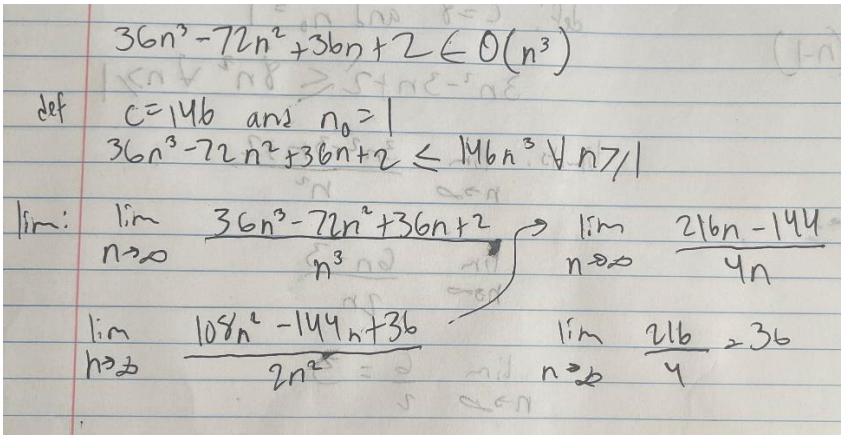
Step Count

```
// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state& before) {

  int count = 0;
  disk_state check = before;

  for(size_t i = 0; i < check.light_count(); i++) {
    if(i % 2 == 0) {  //incrementing
      for(size_t j = 0; j < check.total_count() - 1; j++) {
                    if(check.get(j) > check.get(j + 1)) {
                            check.swap(j);
                            count++;
                    }
            }
    } else {  //not incrementing
      for(size_t j = check.total_count() - 2; j < check.total_count() - 1; j--) {
                    if(check.get(j) > check.get(j + 1)) {
                            check.swap(j);
                            count++;
                    }
            }
    }
      }
      return sorted_disks(disk_state(check), count);
}
```

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

$= 36n^3 - 72n^2 + 36n + 2$

Proof



Alternate

Pseudocode

Count = 0

Disk_state = before

from i to light_count

go left to right

if adjacent disks differ disk_state.get(j) > disk_state.get(j+1)

swap() disk state and increment swap count

Return sorted disks state and swap count

Step Count

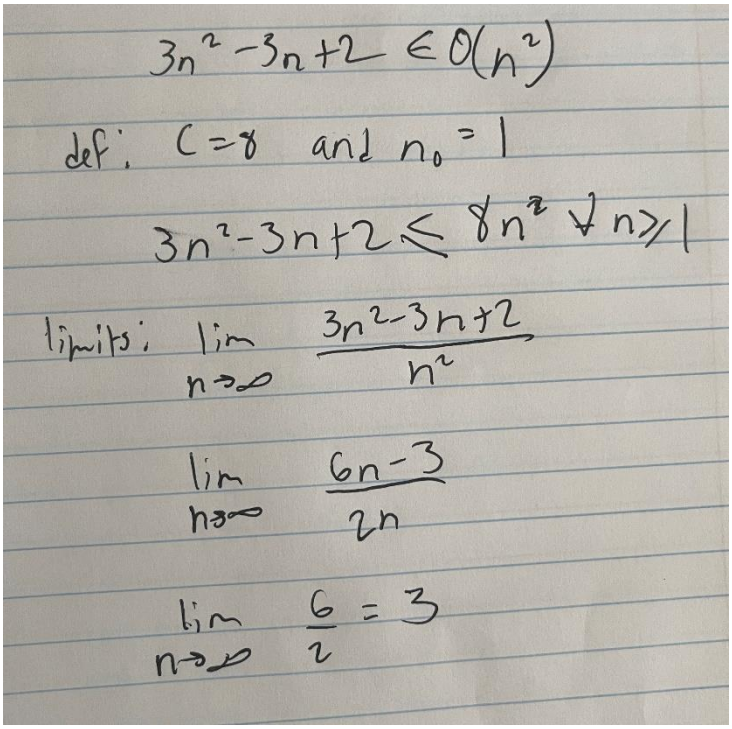
```
// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {

  int count = 0;
  disk_state check = before;

        for(size_t i = 0; i < check.light_count(); i++) {
                for(size_t j = 0; j < check.total_count() - 1; j++) {
                        if(check.get(j) > check.get(j + 1)) {
                                check.swap(j);
                                count++;
                        }
                }
        }
        return sorted_disks(disk_state(check), count);
}
```

n/2 · n − 1 · 2 · 3 · 1

$SC = 2 + (\frac{n}{2}) \times 6(n-1)$

$= 3n^2 - 3n + 2$

Proof


$3n^2 - 3n + 2 \in O(n^2)$
def: $C = 8$ and $n_0 = 1$
$3n^2 - 3n + 2 \leq 8n^2 \; \forall \; n \geq 1$
limits: $\lim_{n \to \infty} \frac{3n^2 - 3n + 2}{n^2}$
$\lim_{n \to \infty} \frac{6n - 3}{2n}$
$\lim_{n \to \infty} \frac{6}{2} = 3$