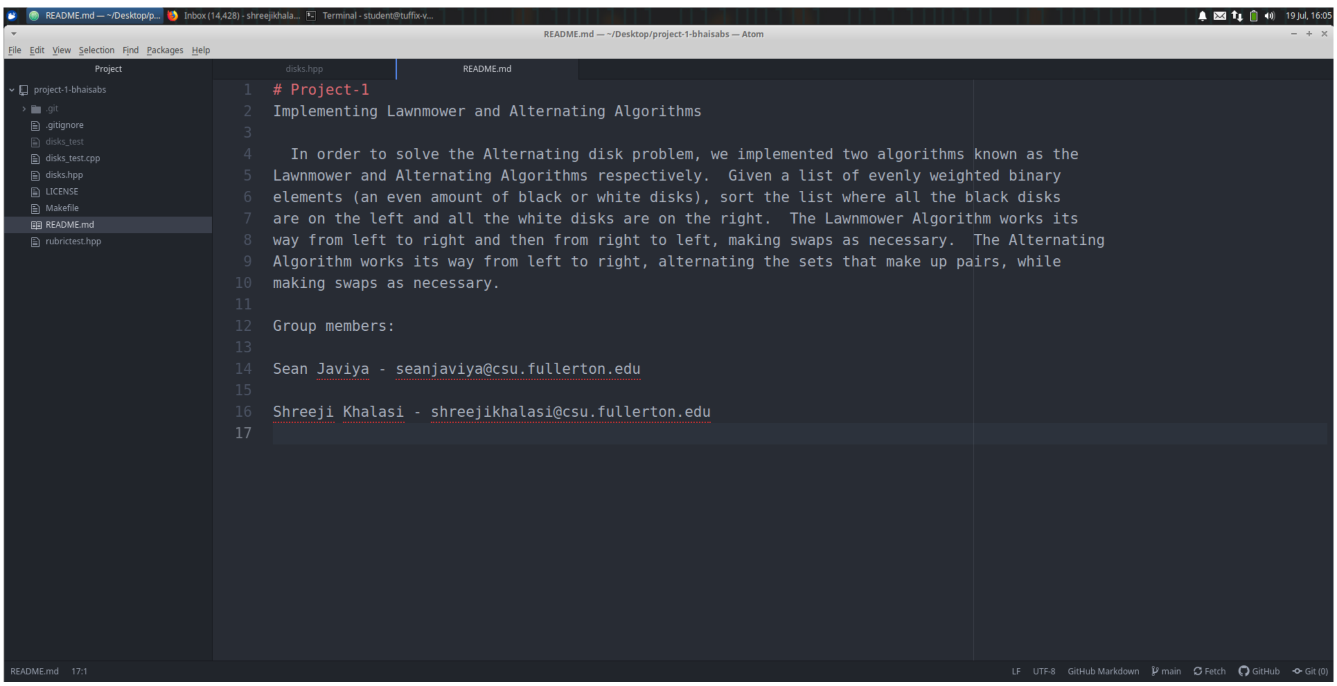
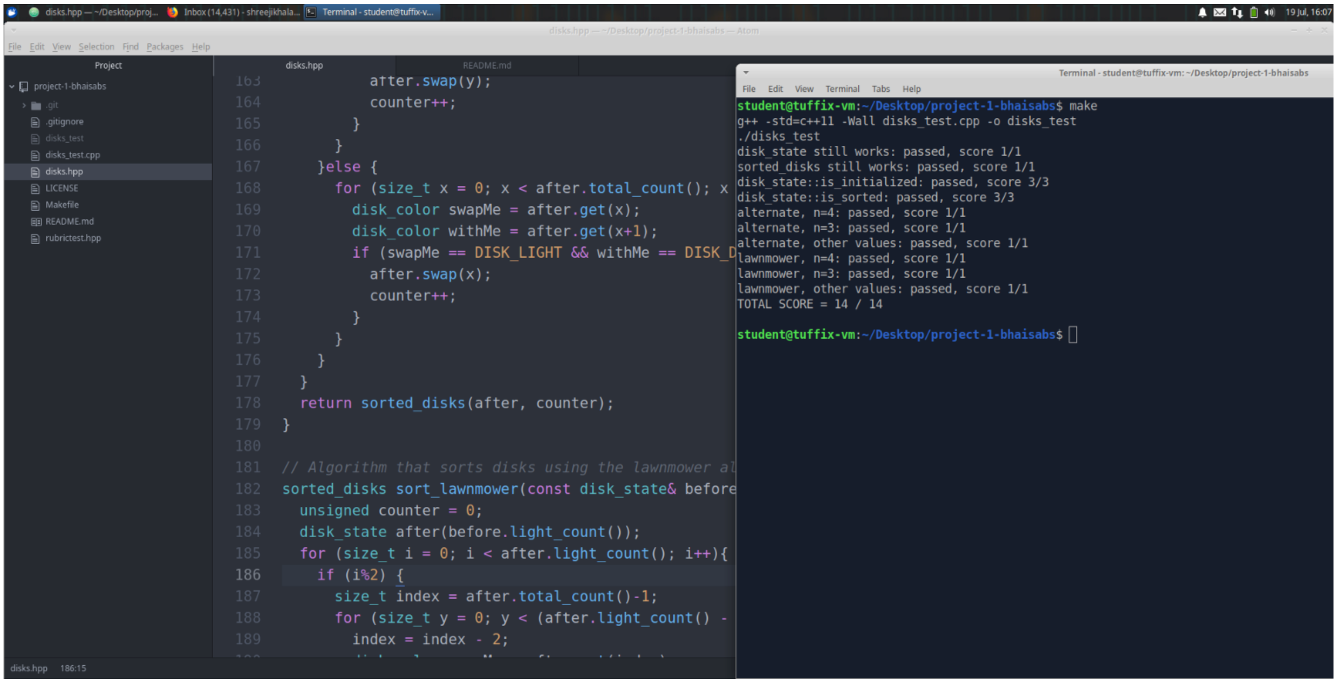
Sean Javiya
Shreeji Khalasi

Project 1

In order to solve the Alternating disk problem, we implemented two algorithms known as the Lawnmower and Alternating Algorithms respectively. Given a list of evenly weighted binary elements (an even amount of black or white disks), sort the list where all the black disks are on the left and all the white disks are on the right. The Lawnmower Algorithm works its way from left to right and then from right to left, making swaps as necessary. The Alternating Algorithm works its way from left to right, alternating the sets that make up pairs, while making swaps, as necessary.

Screenshots:





<u>Pseudocode:</u>

<u>1)</u> Alternate Algorithm

```
sorted_disks sort_alternate(const disk_state& before) {
  counter = 0;
  after = new disk_state;
  for (i=0; i < 2n; i++) {
    if (i is odd) {
      for (y = 1; y < 2n-1; y = y + 2;{
        if (y is DISK_LIGHT and y+1 is DISK_DARK){
          swap(y);
          counter ++
        }
      }
    }
    else {
      for(x = 0; x < 2n; x = x + 2){
        if(x is DISK_LIGHT and x+1 is DISK_DARK){
          swap(x);
          counter ++;
        }
      }
    }
  }
  return sorted_disks(after, counter);
}
```

<u>2)</u> Lawnmower Algorithm

```
sorted_disks sort_lawnmower(const disk_state& before) {
  counter = 0;
```

```
  after = new disk_state;
  for (i = 0; i < n; i++) {
    if (i is odd) {
      index = 2n-1;
      for (y = 0; y < n - 1; y++) {
        index = index - 2;
        if (y is DISK_LIGHT and y+1 is DISK_DARK){
          swap(y);
          counter ++
        }
      }
    }
    else {
      for (x = 0; x < 2n; x = x +2) {
        if (x is DISK_LIGHT and x+1 is DISK_DARK){
          swap(x);
          counter ++
        }
      }
    }
  }
return sorted_disks(after, counter);
}
```

<u>Step Count and Mathematical Analysis</u>

Step Count:

1) 14n^2 + 3n + 1

sorted_disks sort_alternate(const disk_state& before) {
  counter = 0; **step count = 1**
  after = new disk_state; **step count = n (the constructor iterates n times in its for loop)**
  for (i=0; i < 2n; i++) { **step count = 2n * (7n + 1) = 14n^2 + 2n**
    if (i is odd) { **step count = 1 + max(7n - 7, 7n) = 7n + 1**
      for (y = 1; y < 2n-1; y = y + 2;{ **step count = (n - 1) * 7**
        if (y is DISK_LIGHT and y+1 is DISK_DARK){ **step count = 3**
          swap(y); **step count = 3**
          counter ++ **step count = 1**
        }
      }
    }
    else {
      for(x = 0; x < 2n; x = x + 2){ **step count = (n) * 7**
        if(x is DISK_LIGHT and x+1 is DISK_DARK){ **step count = 3**
          swap(x); **step count = 3**
          counter ++; **step count = 1**
        }
      }
    }
  }
  return sorted_disks(after, counter);

}

2) 9n^2 – 4n + 1

```
sorted_disks sort_lawnmower(const disk_state& before) {
  counter = 0;   step count = 1
  after = new disk_state;    step count = n (the constructor iterates n times in its for loop)
  for (i = 0; i < n; i++) {
         step count = (n) * (9n -5) = 9n^2 - 5n
    if (i is odd) {  step count = 1 + max(9n - 6, 7n)   = 9n - 5
      index = 2n-1;  step count = 3
      for (y = 0; y < n - 1; y++) {  step count = (n - 1) * 9 = 9n - 9
        index = index – 2;  step count = 2
        if (y is DISK_LIGHT and y+1 is DISK_DARK){  step count = 3
          swap(y);   step count = 3
          counter ++   step count = 1
        }
      }
    }
    else {
      for (x = 0; x < 2n; x = x +2) {   step count = n * 7 = 7n
        if (x is DISK_LIGHT and x+1 is DISK_DARK){   step count = 3
          swap(x);   step count = 3
          counter ++   step count = 1
        }
      }
    }
  }
return sorted_disks(after, counter);
}
```

Proof and Analysis:

1) (Alternate Algorithm) 14n^2 + 3n + 1 belongs to O(n^2)

$$\lim_{n\to\infty}\frac{F(n)}{G(n)}\,!=\infty$$

let F(n)= 14n^2 + 3n + 1

G(n) = O(n^2)

$$\lim_{n\to\infty}\frac{14n^2+3n+1}{n^2}$$

Using L’Hopital’s Rules

$$\lim_{n\to\infty}\frac{28n+3}{2n}$$

$$14+\lim_{n\to\infty}\frac{3}{2n}$$

$= 14 + 0 = 14$

The limit is defined and non-negative, therefore this algorithm belongs to the order of O(n^2).

2) (Lawnmower Algorithm) 9n^2 – 4n + 1 belongs to O(n^2)

$$\lim_{n \to \infty} \frac{F(n)}{G(n)} != \infty$$

let F(n)= 9n^2 – 4n + 1

G(n) = O(n^2)

$$\lim_{n \to \infty} \frac{9n^2 - 4n + 1}{n^2}$$

Using L'Hopital's Rules

$$\lim_{n \to \infty} \frac{18n - 4}{2n}$$

$$9 + \lim_{n \to \infty} \frac{-4}{2n}$$

$= 9 + 0 = 9$

The limit is defined and non-negative, therefore this algorithm belongs to the order of O(n^2).