Justin Nguyen - justindnguyen03@csu.fullerton.edu

Victor Nguyen - Victor326@csu.fulleron.edu

Himani Tawade

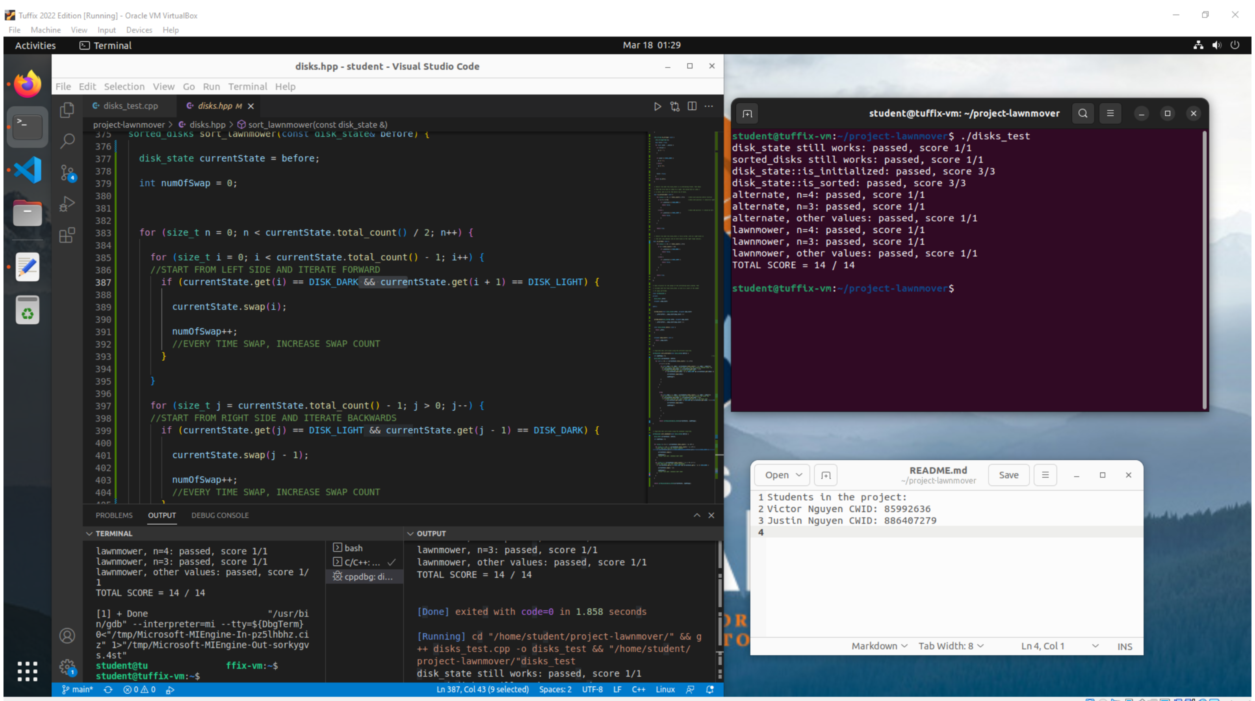
CPSC 335: Algorithm Engineering

19 March 2023

CPSC 335 Project 1: Implementing Algorithms

README.md Screenshot:

Code Compiling and Executing Screenshot:



The Lawnmower Algorithm:

Code:

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

  for (size_t n = 0; n < currentState.total_count() / 2; n++) {
    //START FROM LEFT SIDE AND ITERATE FORWARD
    for (size_t i = 0; i < currentState.total_count() - 1; i++) {
      if (currentState.get(i) == DISK_DARK && currentState.get(i + 1) == DISK_LIGHT) {
        currentState.swap(i);
        numOfSwap++;
      }
    }
    for (size_t j = currentState.total_count() - 1; j > 0; j--) {
      //START FROM RIGHT SIDE AND ITERATE BACKWARDS
      if (currentState.get(j) == DISK_LIGHT && currentState.get(j - 1) == DISK_DARK) {
        currentState.swap(j - 1);
        numOfSwap++;

      }
    }
  }

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

Pseudocode:

```
The pseudocode for lawnmower_sort is:

def lawnmower_sort(disk_state) {
    let n = total length of disk_state
    let swap_count = 0
    for (i from 0 to n / 2, incrementing by 1) {

        //sort starting from left towards right
        for (j from 0 to n-1, incrementing by 1) {
            if (disk at index j == dark && disk at index j+1 == light) {
                swap (disk at index j and disk at index j+1)
                increase swap count by 1
            }
        }

        //sort starting from right towards left
        for (k from n-1 to 0, decrementing by 1) {
            if (disk at index k == light && disk at index k-1 == dark) {
                swap(disk at index k and disk at index k-1)
                increase swap count by 1
            }
        }
    }
    return lawnmower_sort(disk_state, swap count)
}
```

Step Count:

```
The pseudocode for lawnmower_sort is:

def lawnmower_sort(disk_state) {
    let n = total length of disk_state     1 tu
    let swap_count = 0        1 tu
    for (i from 0 to n / 2, incrementing by 1) {     n/2 + 1 times

        //sort starting from left towards right
        for (j from 0 to n-1, incrementing by 1) {   n times
            if (disk at index j == dark && disk at index j+1 == light) {   4 tu
                swap (disk at index j and disk at index j+1)
                increase swap count by 1     1 tu
            }
        }

        //sort starting from right towards left
        for (k from n-1 to 0, decrementing by 1) {   n times
            if (disk at index k == light && disk at index k-1 == dark) {   4 tu
                swap(disk at index k and disk at index k-1)
                increase swap count by 1     1 tu
            }
        }
    }
    return lawnmower_sort(disk_state, swap count)
}
```

Step Count for Lawnmower Algorithm:

$$SC = 1 + 1 + \left(\frac{n}{2}+1\right) \cdot \left[(n)(4+1) + (n)(4+1)\right]$$

$$= 2 + \left(\frac{n}{2}+1\right) \cdot \left[5n + 5n\right]$$

$$= 2 + \left(\frac{n}{2}+1\right) \cdot (10n)$$

$$= 2 + 5n^2 + 10n$$

$$= \boxed{5n^2 + 10n + 2 \quad tu}$$

Prove $5n^2 + 10n + 2 \in O(n^2)$:

By Limit's Theorem,

$$\lim_{n\to\infty} \frac{5n^2+10n+2}{n^2} = \lim_{n\to\infty} 5 + \frac{10}{n} + \frac{2}{n^2} = 5 + 0 + 0 = 5 \geq 0 \text{ and a constant}$$

Therefore,

$$\boxed{5n^2 + 10n + 2 \in O(n^2)}$$

The Alternate Algorithm:

Code:

```
};

// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {
  int numOfSwap = 0;                                              //record # of step swap
  disk_state currentState = before;
   for (int i = 0; i < currentState.total_count() / 2; i++){
      if (i % 2 == 0){
        for (int j = 0; j < currentState.total_count() - 1; j = j+2){
          //FOR LOOP STARTS FROM FIRST DISK FROM LEFT SIDE AND ITERATES EVERY TWO
          if (currentState.get(j) != currentState.get(j + 1)){
            //IF CURRENT DISK DOES NOT EQUAL DISK NEXT TO IT ON ITS RIGHT
               if (currentState.get(j) == DISK_DARK && currentState.get(j + 1) == DISK_LIGHT){
                  currentState.swap(j);
                  numOfSwap++;
          }
          }
        }
      }

      else{
        for (int index = 1; index < currentState.total_count() - 2; index = index+2){
          //FOR LOOP STARTS AT SECOND DISK FROM LEFT SIDE AND ITERATES FOR EVERY TWO
          if (currentState.get(index) != currentState.get(index + 1)){
            //IF CURRENT DISK DOESN'T EQUAL CURRENT DISK NEXT TO IT ON RIGHT SIDE
              if (currentState.get(index) == DISK_DARK && currentState.get(index + 1) == DISK_LIGHT){
                  currentState.swap(index);
                  numOfSwap++;
         }
          }
              }
          }
      }
      return sorted_disks(disk_state(currentState), numOfSwap);
}
```

Pseudocode:

```
The pseudocode for alternate_sort is:

def alternate_sort(disk_state) {
    let n = total length of disk_state
    let swap_count = 0
    for (i from 0 to n / 2, incrementing by 1) {

        //sort starting from leftmost disk
        if (i % 2 == 0) {
            for (j from 0 to n-1, incrementing by 2) {
                if (disk at index j == dark && disk at index j+1 == light) {
                    swap (disk at index j and disk with index j+1)
                    increase swap count by 1
                }
            }
        }

        //sort starting from second leftmost disk
        else {
            for (k from 1 to n-2, incrementing by 2) {
                if (disk at index k == dark && disk at index k+1 == light) {
                    swap (disk at index k and disk with index k+1)
                    increase swap count by 1
                }
            }
        }
    }
    return alternate_sort(sorted disks, swap count)
}
```

Step Count:

```
The pseudocode for alternate_sort is:

def alternate_sort(disk_state) {
    let n = total length of disk_state     1 tu
    let swap_count = 0                1 tu
    for (i from 0 to n / 2, incrementing by 1) {   n/2+1 times

        //sort starting from leftmost disk
        if (i % 2 == 0) {   2 tu
            for (j from 0 to n-1, incrementing by 2) {   (n-1)/2 +1 times
                if (disk at index j == dark && disk at index j+1 == light) {   4 tu
                    swap (disk at index j and disk with index j+1)
                    increase swap count by 1   1 tu
                }
            }
        }

        //sort starting from second leftmost disk
        else {
            for (k from 1 to n-2, incrementing by 2) {   (n-2-1)/2 +1 times
                if (disk at index k == dark && disk at index k+1 == light) {   4 tu
                    swap (disk at index k and disk with index k+1)
                    increase swap count by 1   1 tu
                }
            }
        }
    }
    return alternate_sort(sorted disks, swap count)
}
```

Step Count for Alternate Algorithm:

$$SC = 1 + 1 + \left(\frac{n}{2}+1\right)\cdot\left[2 + \max\left(\left(\frac{n-1}{2}+1\right)\cdot(5)\ ,\ \left(\frac{n-3}{2}+1\right)\cdot(5)\right)\right]$$

$$= 2 + \left(\frac{n}{2}+1\right)\cdot\left[2 + \left(\frac{5n}{2} - \frac{5}{2} + 5\right)\right]$$

$$= 2 + \left(\frac{n}{2}+1\right)\cdot\left[\frac{5n}{2} + \frac{9}{2}\right]$$

$$= 2 + \frac{5n^2}{4} + \frac{9}{4} + \frac{5n}{2} + \frac{9}{2}$$

$$= \frac{5}{4}n^2 + \frac{5n}{2} + \frac{35}{4}$$

Prove $\frac{5}{4}n^2 + \frac{5}{2}n + \frac{35}{4} \in O(n^2)$:

By Limit's Theorem,

$$\lim_{n\to\infty} \frac{\frac{5}{4}n^2 + \frac{5}{2}n + \frac{35}{4}}{n^2} = \lim_{n\to\infty} \frac{5}{4} + \frac{5}{2n} + \frac{35}{4n^2} = \frac{5}{4} + 0 + 0 = \frac{5}{4} \geq 0 \text{ and a constant}$$

Therefore,

$$\boxed{\frac{5}{4}n^2 + \frac{5}{2}n + \frac{35}{4} \in O(n^2)}$$