Benjamin Yang

Gurman Gill

CPSC 335
Summer 2023

Pseudocode for the Algorithms

```
function sort_alternate(before):
  after = copy(before) // n
  swap_count = 0  // 1
  swap_prior = true  // 1
```

$n+2 = n$

```
  while swap_prior:      // 2  because of the constant iterating once
    swap_prior = false  //1

    for i = 0 to after.total_count() - 2 by 2:  // (n/2 - 1) = n
      if after.get(i) is DISK_LIGHT and after.get(i + 1) is DISK_DARK:  //1
        after.swap(i)  // 1
        swap_count = swap_count + 1  //1
        swap_prior = true  //1
        break  //1
```

n

```
    if not swap_prior:  // 1
      for i = 1 to after.total_count() - 2 by 2:  // (n/2) = n
        if after.get(i) is DISK_LIGHT and after.get(i + 1) is DISK_DARK:  //1
          after.swap(i)  // 1
          swap_count = swap_count + 1  //1
          swap_prior = true  //1
          break  //1
```

n

```
  return sorted_disks(after, swap_count)  //1
```

Time complexity : O(n^2)

```
function sort_lawnmower(before):
  after = copy(before)   // n
  swap_count = 0   // 1
  swap_prior = false  // 1
```

$n+2 = n$

```
  do:
    swap_prior = false   // 1
```

```
    for i = 0 to after.total_count() - 2:   // 2
      if after.get(i) is DISK_LIGHT and after.get(i + 1) is DISK_DARK:   // (n+1)
        after.swap(i)      // 1
        swap_count = swap_count + 1     // 1
        swap_prior = true         // 1

    for i = after.total_count() - 2 down to 0:   // (n+1)
      if after.get(i) is DISK_LIGHT and after.get(i + 1) is DISK_DARK:   // (n+1)
        after.swap(i)    // 1
        swap_count = swap_count + 1       // 1
        swap_prior = true

  while swap_prior   // 1

  return sorted_disks(after, swap_count)     // 1
```

Time Complexity: O(n^2)

<u>Step count for the Algorithms</u>

```
//Alternate
sorted_disks sort_alternate(const disk_state& before) {
  disk_state after(before);   // 1
  unsigned swap_count = 0; //1 step
  bool swap_prior = true; //1 step        3

  while (swap_prior) {   // 1
    swap_prior = false; // 1 step

    for (size_t i = 0; i < after.total_count() - 1; i += 2) //4 step   // (n-1)/2 +1
 {
      if (after.get(i) == DISK_LIGHT && after.get(i + 1) == DISK_DARK) // 4 steps
{
        after.swap(i);   //1
        swap_count++; //1 step
        swap_prior = true; //1 step
        break;  //After a swap, exit the loop
      }
    }
```

```
  if (!swap_prior)  // 1
{
    for (size_t i = 1; i < after.total_count() - 1; i += 2) // 4 steps  // n/2 + 1
 {
      if (after.get(i) == DISK_LIGHT && after.get(i + 1) == DISK_DARK) //4 steps
{
        after.swap(i);  // 1
        Swap_count++; //1 step
        swap_prior = true; //1step
        break;  //After a swap, exit the loop
      }
    }
  }
 }

 return sorted_disks(after, swap_count);  // 1
}
```

7max

Proof that 7n^2+21n+16 belongs to O(n^2)

$$\frac{7n^2+21n+16}{n^2}$$

Take limit

$$\frac{14n + 21}{2n} \rightarrow \frac{14}{2} = 7$$

Since the limit is a constant and it is positive, we know that 7n^2+21n+16 belongs to O(n^2)

Step Count: $7n^2+21n+16$

```
//Lawnmower
sorted_disks sort_lawnmower(const disk_state& before) {
 disk_state after(before);  // 1
  size_t swap_count = 0; // 1 step
  bool swap_prior;  // 1
  do {
    swap_prior = false; //1 step
    for (size_t i = 0; i < after.total_count() - 1; i++)// 4 steps    (n-1)/2 + 1
 {
      if (after.get(i) == DISK_LIGHT && after.get(i + 1) == DISK_DARK) // 4 steps
{
        after.swap(i); // 1
        swap_count++; //1 step
        swap_prior = true; // 1 step
      }
    }
    for (size_t i = after.total_count() - 2; i > 0; i--) //4 steps    n-2
```

3

7

```
{
        if (after.get(i) == DISK_LIGHT && after.get(i + 1) == DISK_DARK) // 4 steps
{
            after.swap(i);  // 1
            Swap_count++; // 1 step
            swap_prior = true; 1 step
        }
    }
  } while (swap_prior);
  return sorted_disks(after, swap_count); // 1
}
```

7max

Step Count: $(49n^2 - 49n - 96)/2$

Proof that (49n^2+49n-96)/2 belongs to O(n^2)

$$\lim_{x \to \infty} \frac{(49/2)n^2 - (49/2)n - 48}{n^2}$$

Take limit

$$\frac{49n + 49/2}{2n} \to \frac{49}{2}$$

Since the limit is a constant and it is positive, we know that (49n^2+49n-96)/2 belongs to O(n^2)

Screenshot Readme

README

File Edit View

```
# 335-project-1-starter
Alternating disks

Group members:
Benjamin Yang eternaldj4@csu.fullerton.edu
Gurman Gill gurman662@csu.fullerton.edu
```