Erine Estrella
Doina Bein
CPSC 335 Algorithm Engineering
M/W 4:00pm - 5:00pm

Project 1 : Pseudocode

The Left to Right Algorithm

Input: An unsorted array (before) the size of 2n
Output: A fully sorted swapped array (before) with the number of swaps that occurred (numOfSwaps)

```
// Algorithm that sorts disks using the left-to-right algorithm.
// disk_state: The state of one row of disks, essentially the array
sorted_disks sort_left_to_right(const disk_state& before) {

int numOfPasses = 0;
int index = 0;
int numOfSwaps =0;

for numOfPasses between 0 to 2n-1[redacted]:
  // The last element at index should be in proper order
  for index between 0 to 2n-numOfPasses -1:
      if before[index] is out of order from before[index+1] then:
        add the numOfSwaps by one
        swap(index)

//returns the "row of disks", the amount of swaps that occured
 return sorted_disks(before,  numOfSwaps);
}
```

The Lawnmower Algorithm

Input: An unsorted array (before) the size of 2n
Output: A fully sorted swapped array (before) with the number of swaps that occurred (numOfSwaps)

```
// Algorithm that sorts disks using the Lawnmower Algorithm
//disk_state: The state of the single row array of disks
sorted_disks sort_lawnmower(const disk_state& before) {

int numOfPasses = 0;
int indexer = 0;
int numOfSwaps =0;

for numOfPasses between 0 to 2n-1:

int  secondIndexer = 2n-1; // The secondIndexer will always start at the end of the array
        // If the numOfPasses is even, then the swaps will go LEFT to RIGHT
        if  numOfPasses%2 == 0:
          // The last element at index should be in proper order
          for indexer between 0 to 2n-numOfPasses-1:
              if before[indexer] is out of order from before[indexer+1] then:
                add the numOfSwaps by one
                swap(indexer)

        // For every other pass through, the numOfPasses will be odd and therefore will be at the rhs of the
        // array. If the numOfPasses is odd, then the swaps will go RIGHT to LEFT
        else:
          // From Right to Left
          for secondIndexer between 2n-numOfPasses-1 to 0:
              if before[secondIndexer] is out of order from before[secondIndexer-1] then:
                add the numOfSwaps by one
                swap(secondIndexer-1)
 return sorted_disks(before, numOfSwaps);
}
```

Project 1 : Mathematical Analysis - Calculating Step Count and Time Complexity

The Left to Right Algorithm

Note: For simplicity with calculations, I have changed the variable names within my code.

Input: An unsorted array (before) the size of 2n

Output: A fully sorted swapped array (before) with the number of swaps that occurred (c)

```
sorted_disks sort_left_to_right(const disk_state& before) {

int i = 0;   (1)
int k = 0;   (1)
int  c = 0;  (1)

for (i = 0; i < 2n-1;i++){ ((2n-1)-0+1) = (2n)

   for (k = 0; k < 2n - i - 1; k++){ ((2n - i -1) - 0 +1) = (2n-i)

        if (before.get(k) == DISK_DARK && before.get(k+1) == DISK_LIGHT){ (3)+3 = 6

          c++;   (2)
          swap(k); (1)
          }
   }
}
 return sorted_disks(before,  c);
}
```

$= (6)(2n - i)$

$= 12n - 6i$

$= \sum_{i=0}^{2n} 12n - 6i \Rightarrow 6\sum_{i=0}^{2n} 2n - i$

$= 6\sum_{i=0}^{2n} 2n - 6\sum_{i=0}^{2n} i$

$= 6\sum_{i=0}^{2n} 2n - 6\left(\frac{2n(2n+1)}{2}\right)$

$= 6\sum_{i=0}^{2n} 2n - 6\left(\frac{4n^2+2n}{2}\right)$

$= 6(2n+1)(2n) - 6(2n+n)$

$= 6(4n^2+2n) - 12n + 6n$

$= 24n^2 + 12n - 12n - 6n$

$= 24n^2 - 6n$

To prove $24n^2 - 6n \in O(n^2)$

$c = 30$

$24n^2 - 6n \leq 30n^2 \quad \forall n \geq n_0$

$-30n^2 \qquad -30n^2$

$30n^2 - 24 + 6n \leq 0$

$6n^2 + 6n \leq 0$ ✓

The Lawnmower Algorithm

Input: An unsorted array (before) the size of 2n

Output: A fully sorted swapped array (before) with the number of swaps that occurred (c)

```
sorted_disks sort_lawnmower(const disk_state& before) {

int i = 0;
int k = 0;
int c =0;

for (i = 0; i < 2n-1;i++){

  int  j  = 2n-1;

  if (i%2 == 0){
      for (k = 0; k < 2n - i - 1; k++){

        if (before.get(k) == DISK_DARK && before.get(k+1) == DISK_LIGHT){
          c++;
          swap(k);
        }
      }

      else{
        for (j; 2n - i - 1 < j ; j--){
            if (before.get(j) == DISK_LIGHT && before.get(j-1) == DISK_DARK){
              c++;
              swap(j-1);
            }
        }
      }
}
 return sorted_disks(before, c);
}
```

(1) (1) (1)

((2n - 1) - 0 + 1) = 2n

(2)

(2)

((2n - i - 1) - 0 + 1) = (2n - i)

S1 is longer

(3) + 3 = 6

(2) (1)) 3

= 6 (2n - i)

= (12n - 6i)

⇒ 2 + 12n - 6i = S1

= ((2n - i - 1) - (2n - 1) + 1)

= (2n - i - 2n + 1)

= (- i + 1) ⇒ (i - 1)

S2

(3) + 3 = 6

(2) (1)) 3

= (6) (i - 1)

= (6i - 6)

= 2 + 6i - 6

⇒ 6i - 4 = S2

cont on next page...

Since the if/else statement takes into consideration the larger SC of S1 & S2, we only use S1.

$$= \sum_{i=0}^{2n} 2 + 12n - 6i$$

$$= 2\sum_{i=0}^{2n} 1 + 2\sum_{i=0}^{2n} 6n - 2\sum_{i=0}^{2n} 3i$$

$$= 2\sum_{i=0}^{2n} 1 + 6\sum_{i=0}^{2n} 2n - 6\left(\frac{2n(2n+1)}{2}\right)$$

$$= 2(2n+1) + 6(2n+1)(2n) - 6\left(\frac{4n^2+2n}{2}\right)$$

$$= 4n+2 + 6(4n^2+2n) - 6(2n+n)$$

$$= 4n+2 + 24n^2+12n - (12n-6n)$$

$$= 4n+2+24n^2+12n-6n$$

$$= 24n^2+4n+12n-6n+2$$

$$= 24n^2+10n+2$$

to prove $24n^2 + 10n + 2 \in O(n^2)$

$$c = 34$$

$$= 24n^2+10n+2 \leq 34n^2 \quad \forall n \geq n_0$$

$$-34n^2 \qquad -34n^2$$

$$= 34n^2-24n^2-10n-2 \leq 0$$

$$= 10n^2-10n-2 \leq 0 \checkmark$$