THE INSERTION SORT ALGORITHM

ALEX GOODSON

CS 131

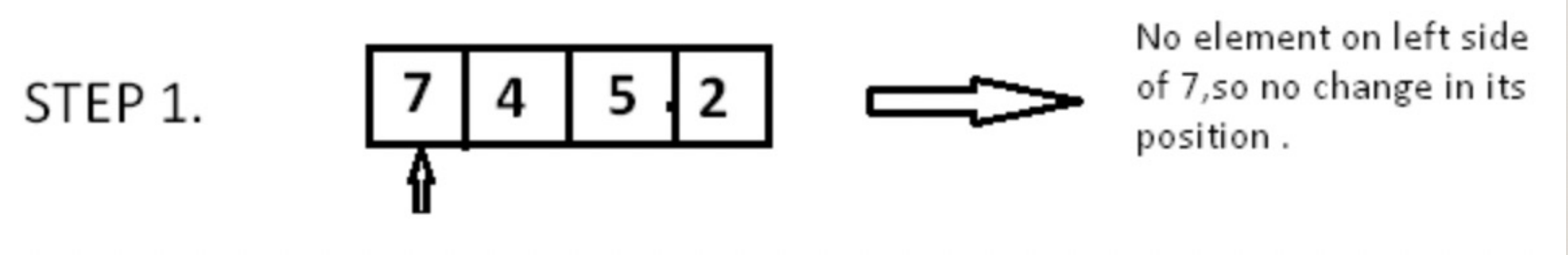
INSERTION SORT AS A PROCEDURE

ALGORITHM 5 The Insertion Sort.

procedure *insertion sort*($a_1, a_2, \ldots, a_n$: real numbers with $n \geq 2$)
for $j := 2$ **to** n
    $i := 1$
    while $a_j > a_i$
        $i := i + 1$
    $m := a_j$
    for $k := 0$ **to** $j - i - 1$
        $a_{j-k} := a_{j-k-1}$
    $a_i := m$
{$a_1, \ldots, a_n$ is in increasing order}

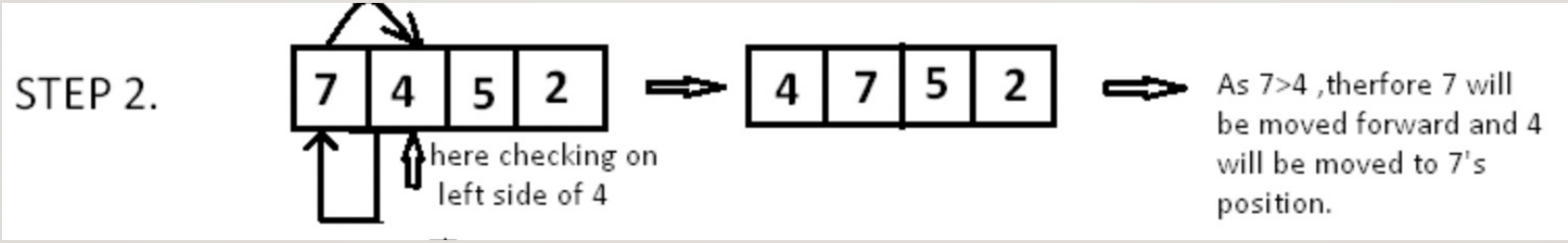
INSERTION SORT ALGORITHM EXPLAINED

- Purpose of insertion sort is to sort array in increasing order
- When using the insertion sort algorithm, we check to see if each element is in the correct order until we reach the current element
- 1st step illustrated below



NEXT STEPS

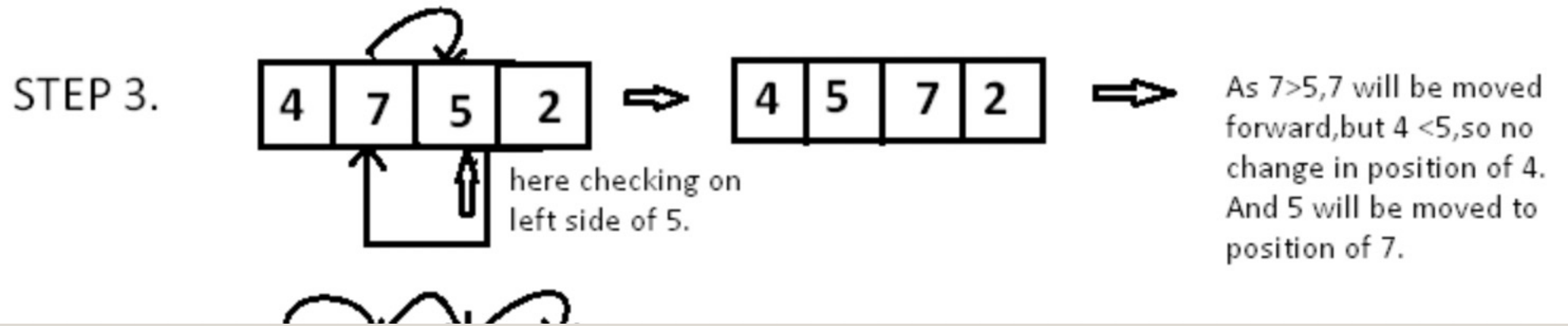
- Each item is compared to the item to its left (works left to right)
- As illustrated below, 4 is compared to 7, and as 7 is larger than 4, it will be moved to the right
- Now the items we have seen thus far in the array (4, and 7) have been sorted



NEXT STEPS

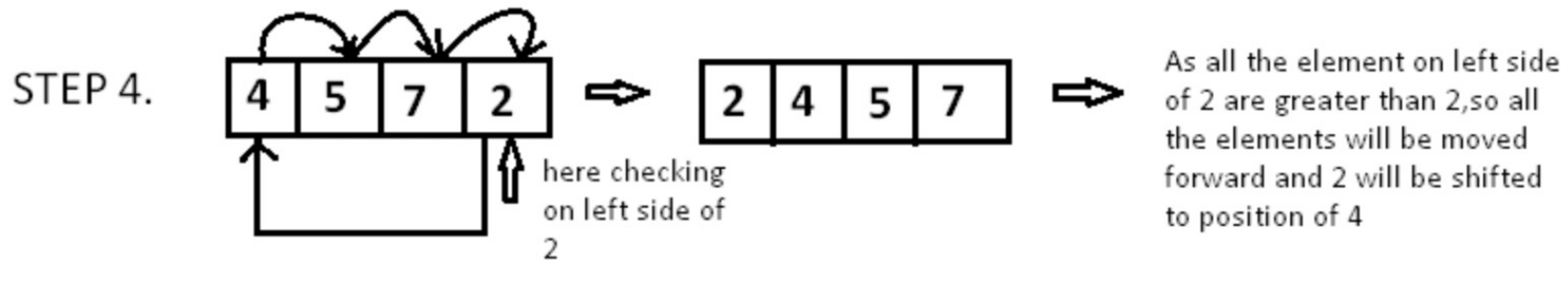
- Next, we repeat the previous step and check on the left side of 5.

- Since 7 is greater than 5, 7 will be moved to the right, but 4 is less than 5 so it will not change in position. 5 will trade places with 7 as illustrated below



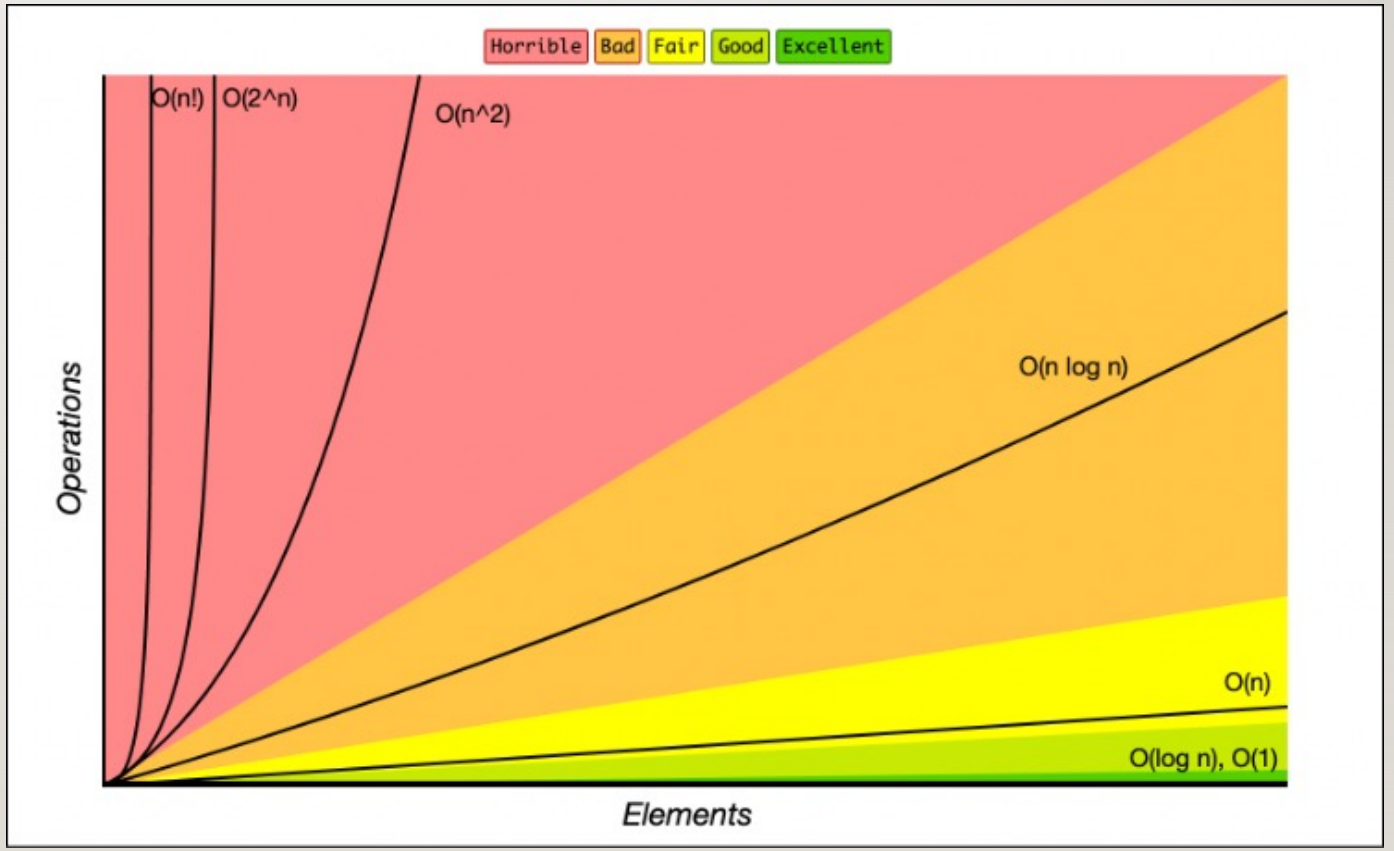
NEXT STEPS

- The process is repeated until the array is completely sorted
- As 2 is less than 7,5, and 4, it is moved to the far left, making it the first number in our array



TIME COMPLEXITY

- Best Case: O(n)
- Average Case: O(n^2)
- Worst Case: O(n^2)



USE CASES FOR INSERTION SORT

- Insertion sort performs best in arrays where most elements are already sorted
- Also performs well in small data sets

THANK YOU