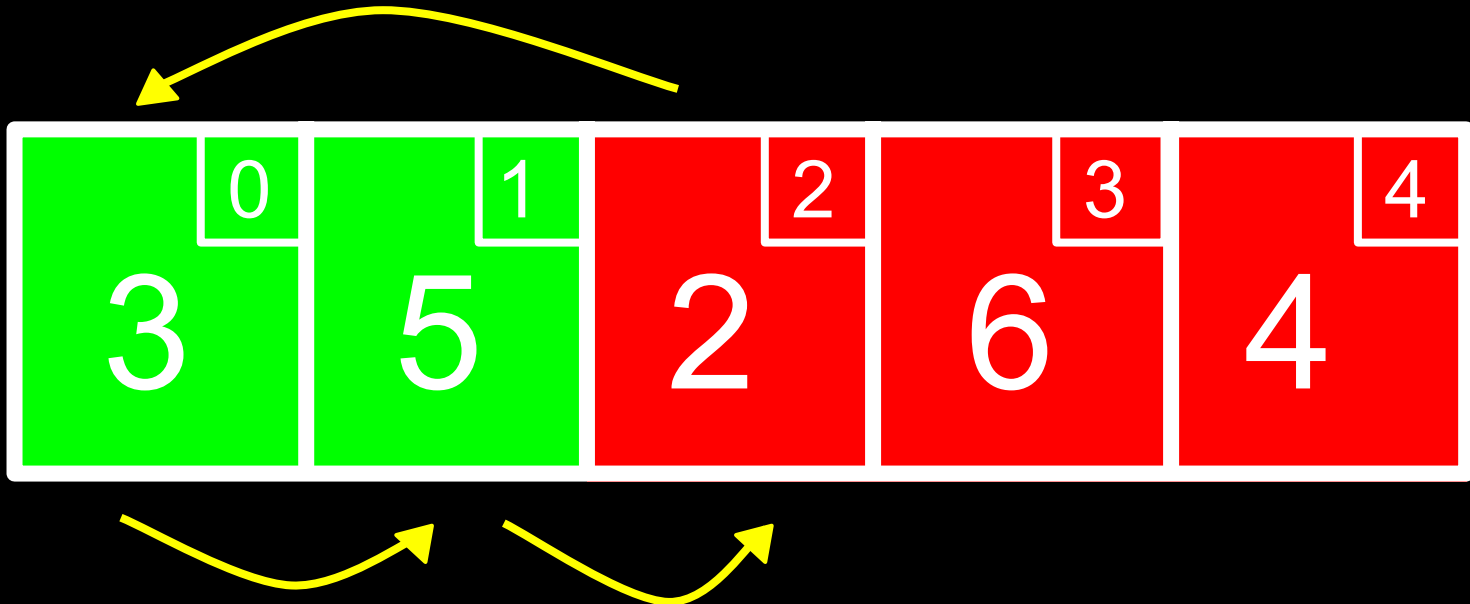


Insertion Sort

Sorted

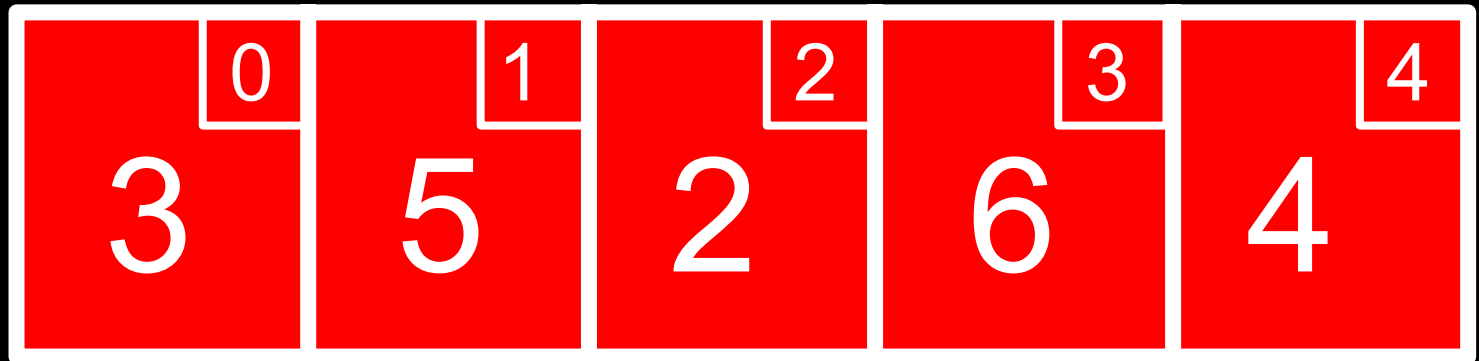
Unsorted



All values start as **Unsorted**

Sorted

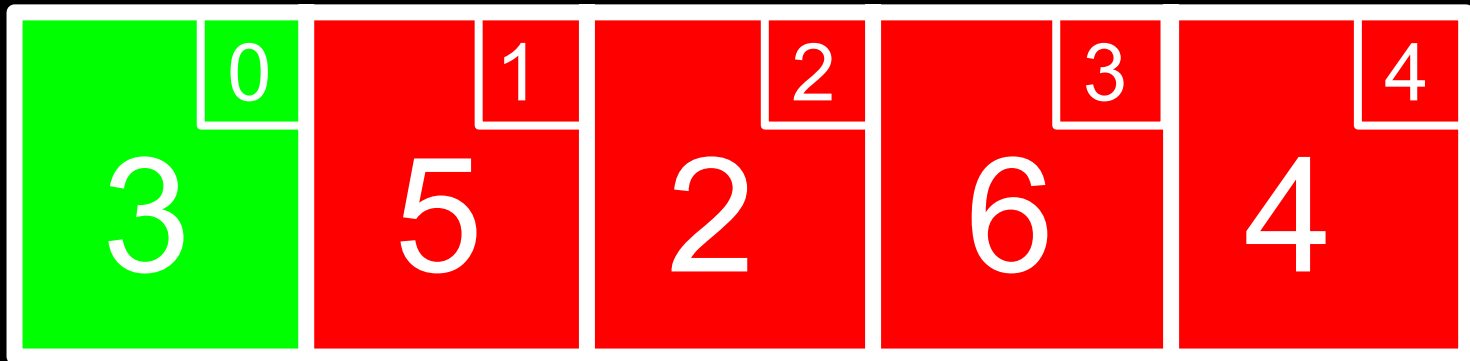
Unsorted



Add first value to **Sorted**

Sorted

Unsorted

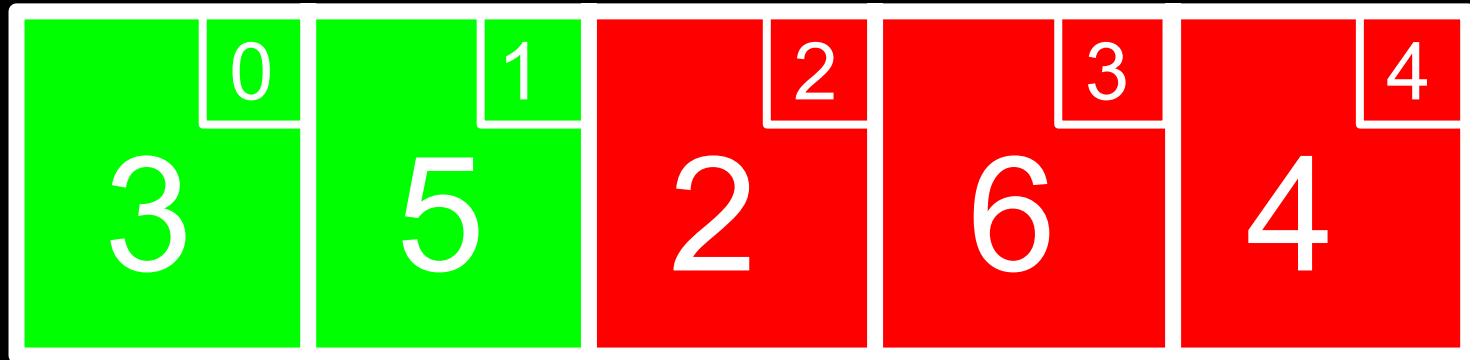


$5 > 3$

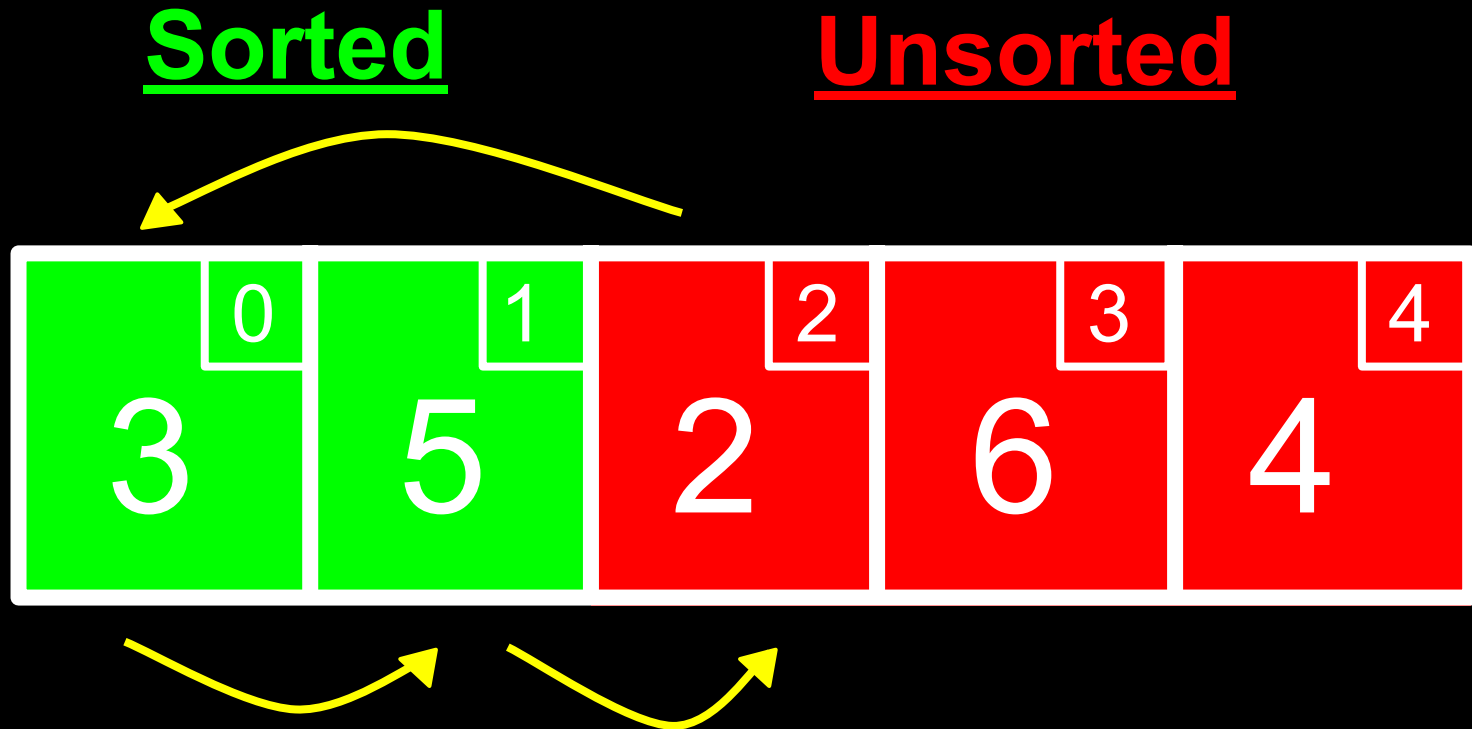
insert 5 to right of 3

Sorted

Unsorted



$2 < 5$ and $2 < 3$
shift 3 and 5
insert 2 to left of 3

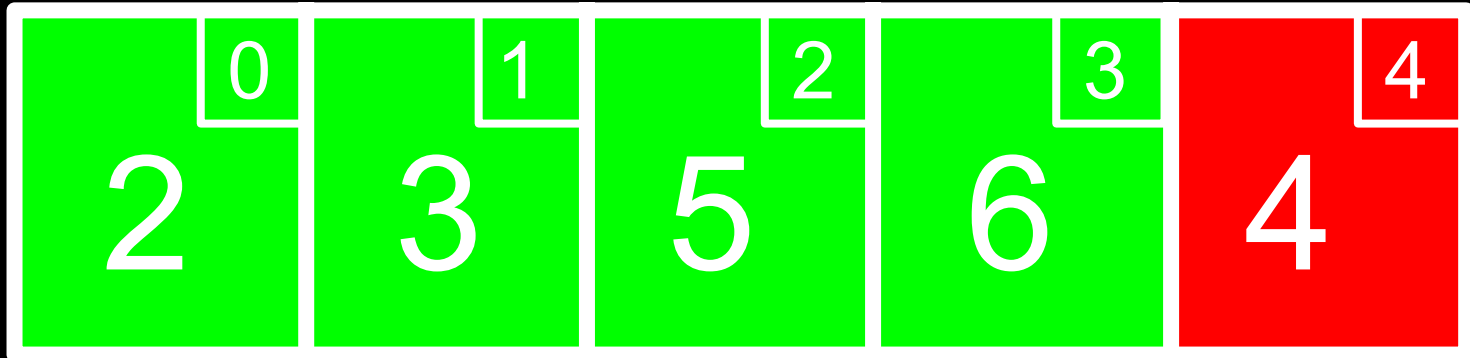


$6 > 5$

insert 6 to right of 5

Sorted

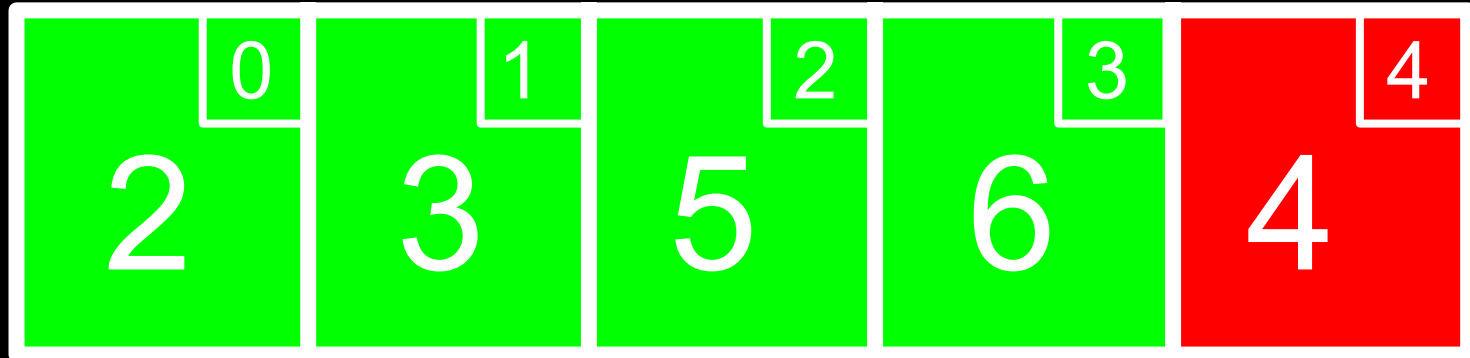
Unsorted



$4 < 6$, $4 < 5$, and $4 > 3$
shift 5 and 6
insert 4 to right of 3

Sorted

Unsorted



For each unsorted element n :

1. Determine where in sorted portion of the list to insert n

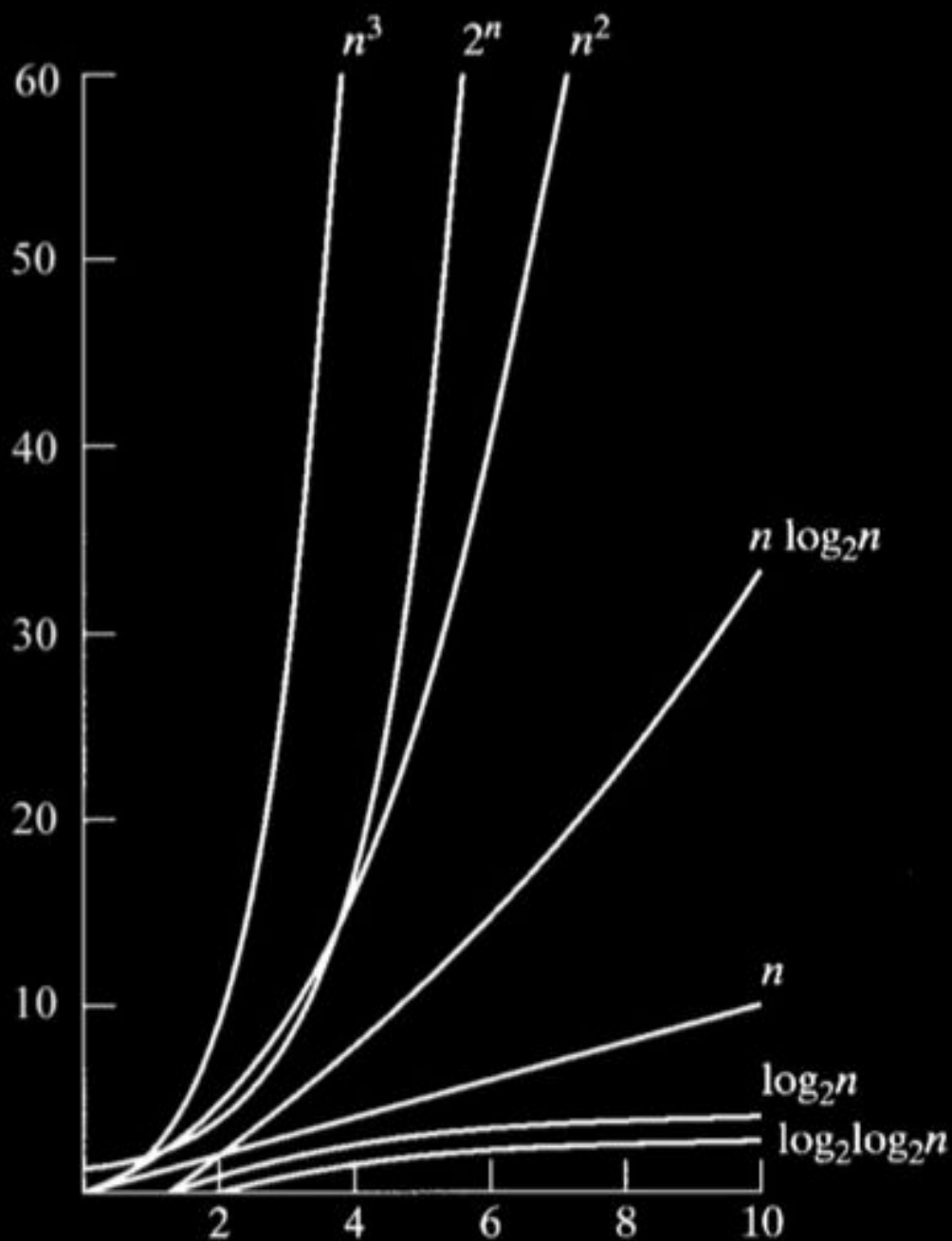
2. Shift sorted elements rightwards as necessary to make room for n

3. Insert n into sorted portion of the list


```
for i = 0 to n - 1  
  element = array[i]  
  j = i  
  while (j > 0 and array[j - 1] > element)  
    array[j] = array[j - 1]  
    j = j - 1  
  array[j] = element
```

What's the worst case runtime of insertion sort?

What's the best case runtime of insertion sort?



	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
O	n^2	n^2	n^2	$n \log n$
Ω	n	n^2	n	$n \log n$
Θ		n^2		$n \log n$