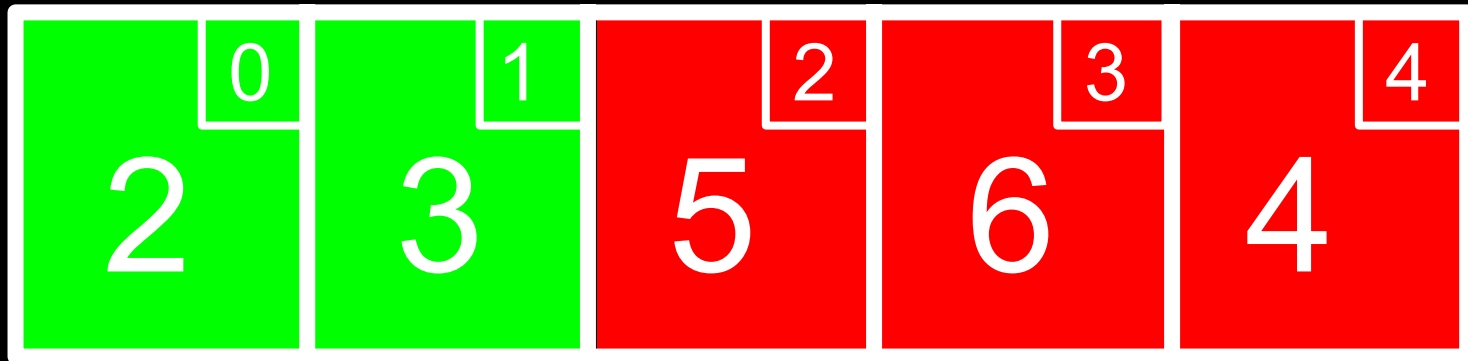


Selection Sort

Sorted

Unsorted



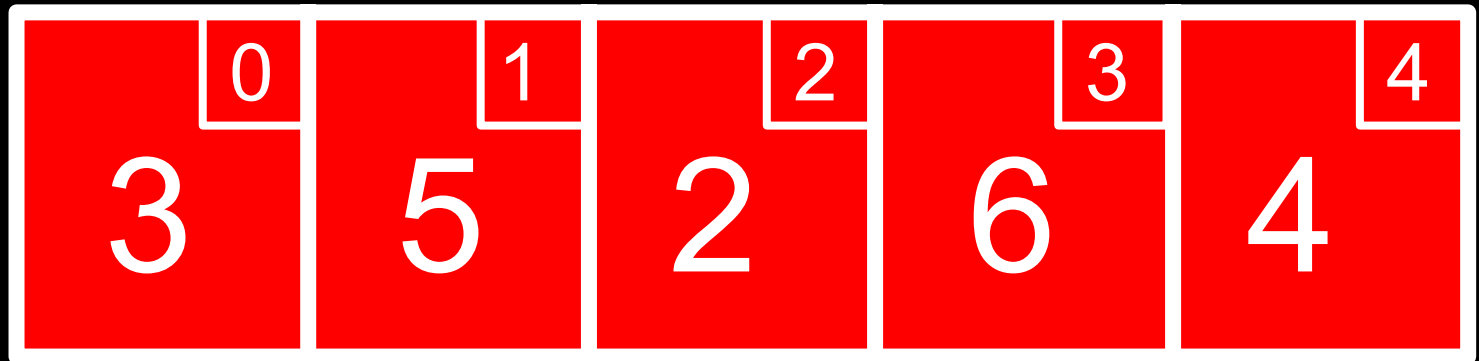
Algorithm

- 1. Find the smallest unsorted value**
- 2. Swap that value with the first unsorted value**
- 3. Repeat from Step 1 if there are still unsorted items**

All values start as **Unsorted**

Sorted

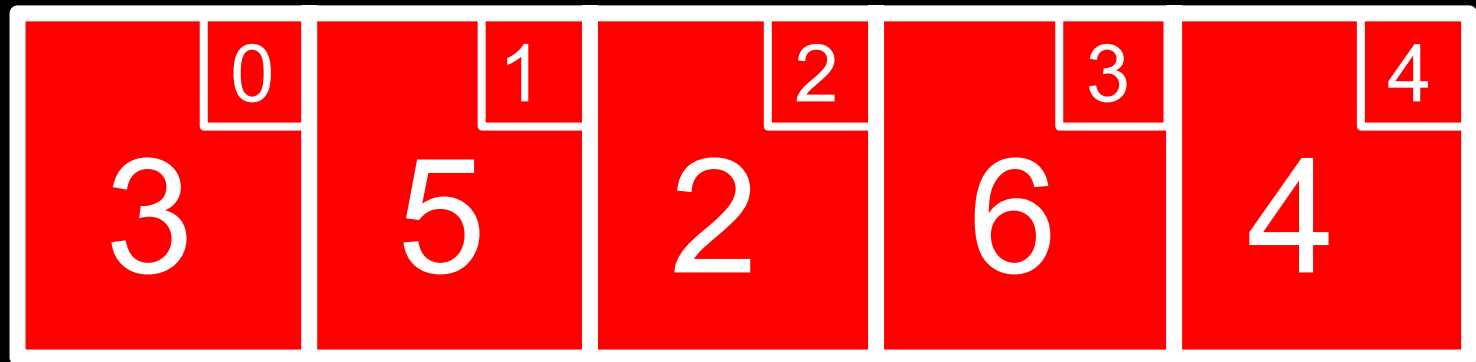
Unsorted



**First pass:
2 is smallest, swap with 3**

Sorted

Unsorted

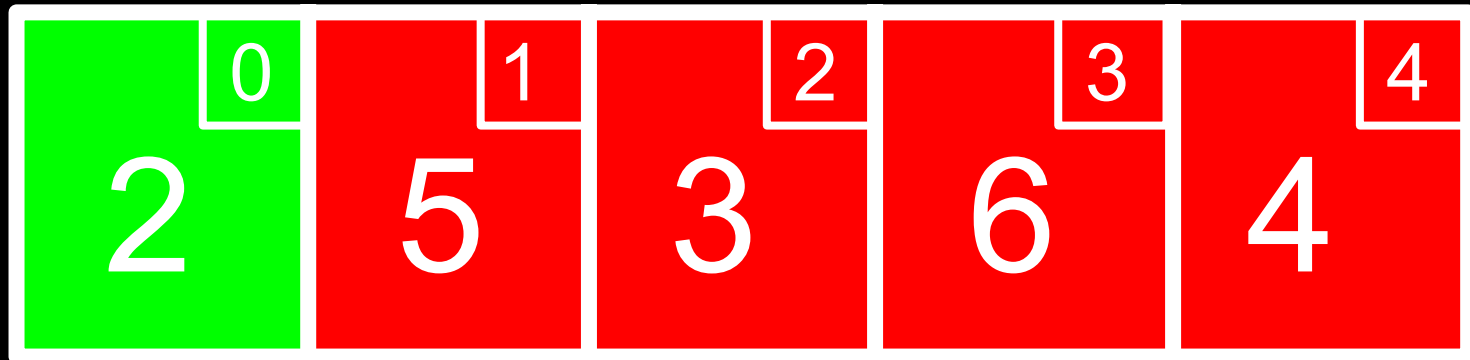


Swap

**Second pass:
3 is smallest, swap with 5**

Sorted

Unsorted

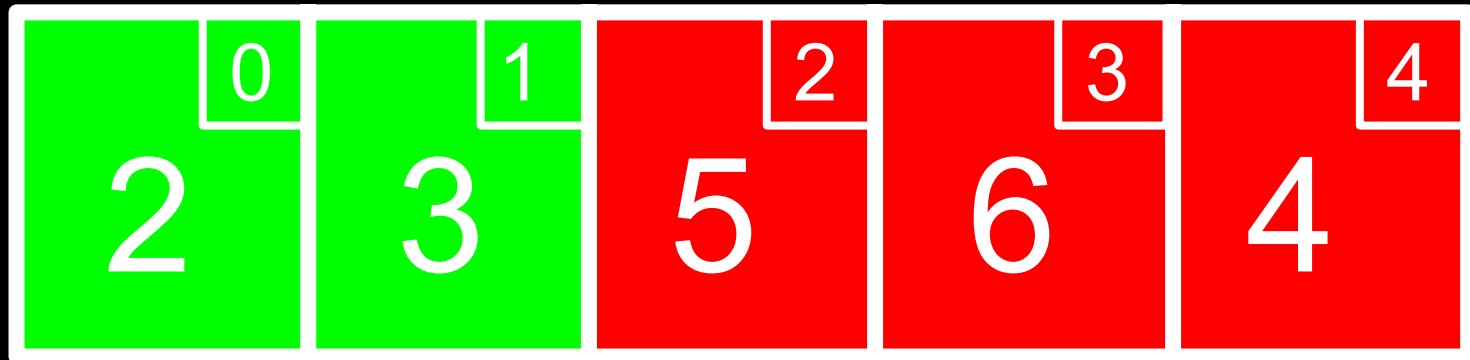


Swap

**Third pass:
4 is smallest, swap with 5**

Sorted

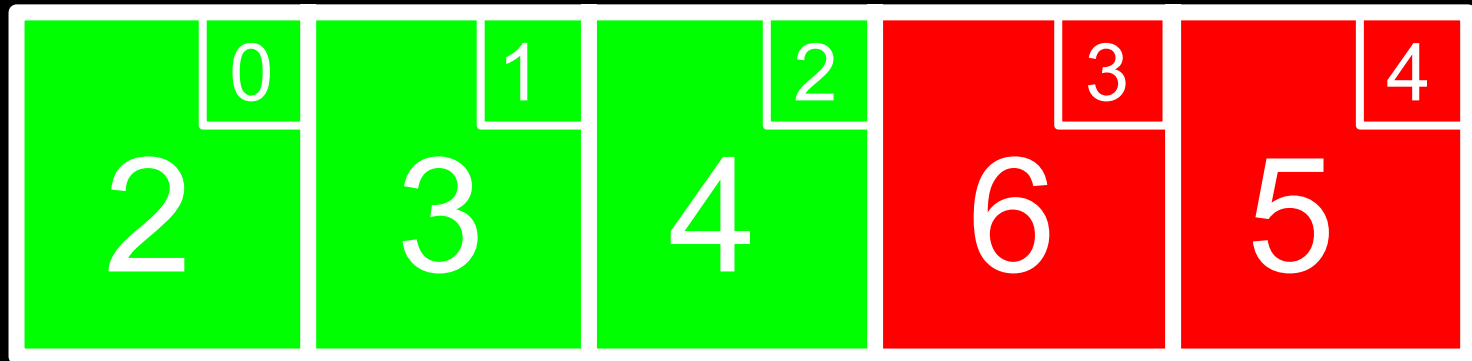
Unsorted

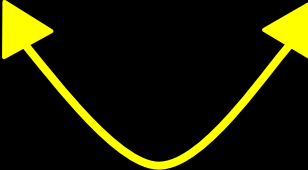


**Fourth pass:
5 is smallest, swap with 6**

Sorted

Unsorted

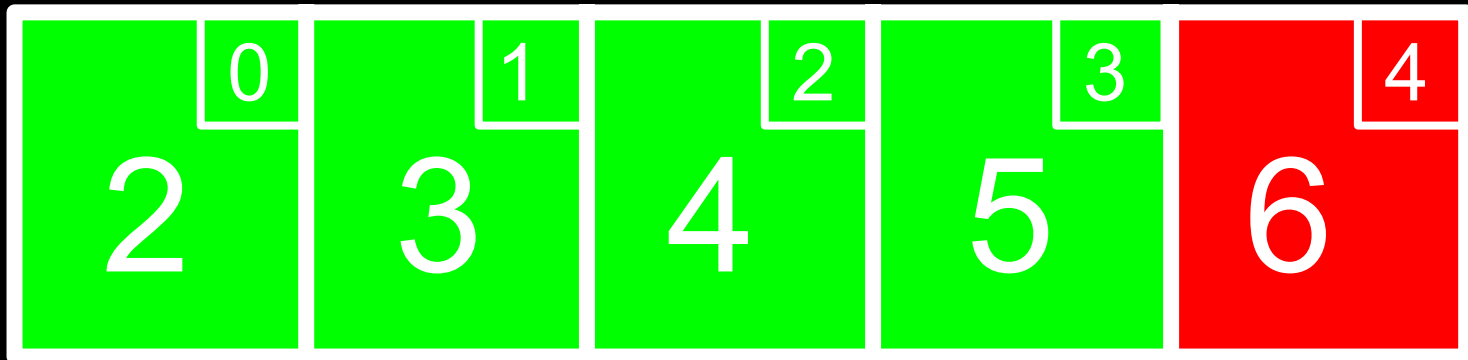


Swap 

**Fifth pass:
6 is the only value left, done!**

Sorted

Unsorted

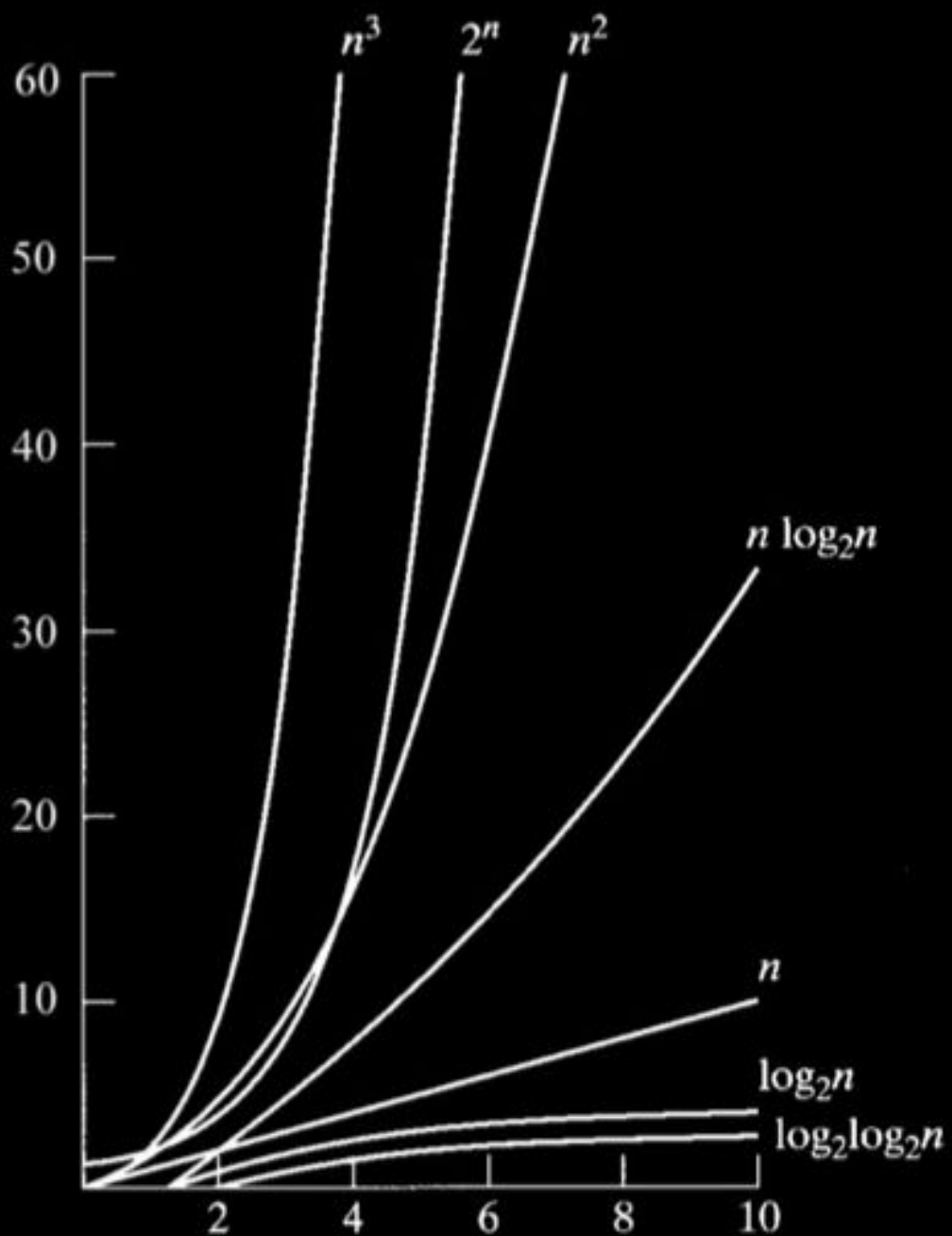



```
for i = 0 to n - 2  
  min = i  
  for j = i + 1 to n - 1  
    if array[j] < array[min]  
      min = j;  
  if min != i  
    swap array[min] and array[i]
```

What's the best case runtime of selection sort?

What's the worst case runtime of selection sort?

What's the expected runtime of selection 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		$n \log n$