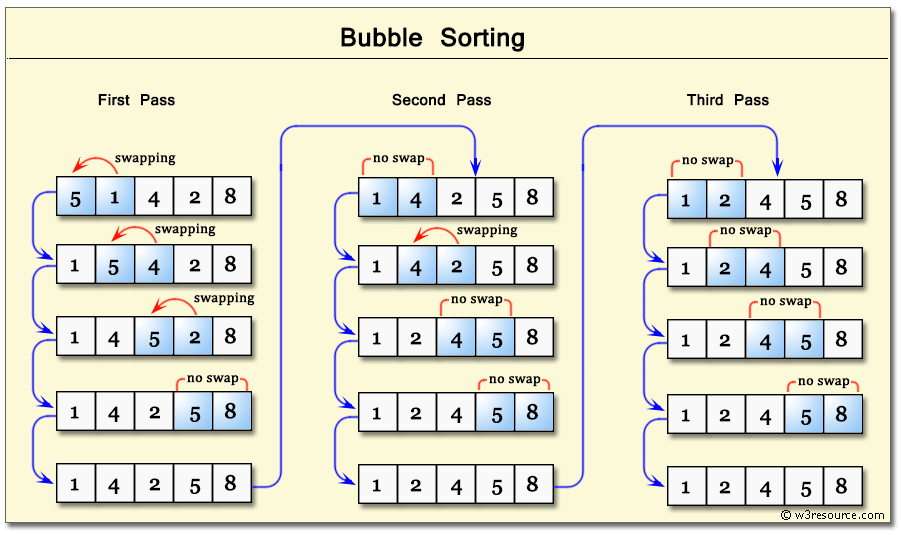
# Bubble Sort

We go through the array, swapping index and index+1 if index+1 is bigger than index.  
We do multiple passes.  
We keep track in a bool whether we made any swap in the current pass.  
If we didn’t make any swap in the current swap, we are done.





# Selection sort

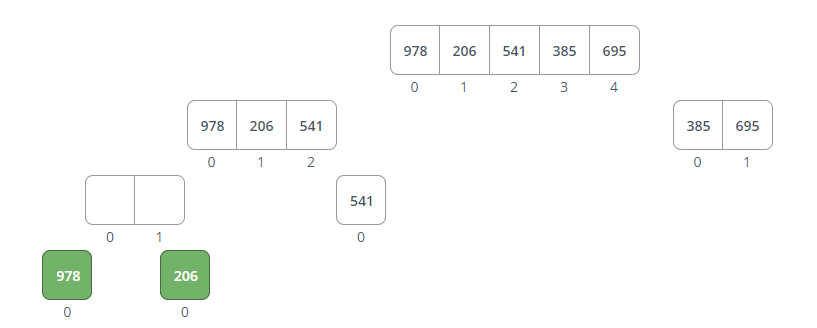
Grab the min. value of the array, put it on 1st place  
Then check n-1, grab the new low, place it to 2nd place  
Then check n-2, grab the new low, place it to 3rd place

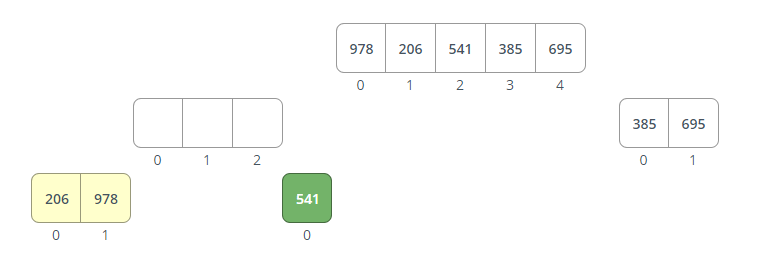
# Insertion sort

Start on 2nd item, compare it to 1st 🡪 if lower, swap  
Start on 3rd item, compare it to 2nd 🡪 if lower, swap 🡪 compare it to 1st 🡪 if lower, swap

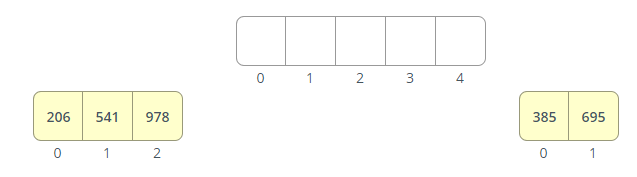
Start on **n, compare it to n-1, swap, then n-2, swap**

# Merge sort

Split array into 2 equal parts  
Select the left subarray, split it again into 2 equal parts  
Select the left subarray, split it again into 2 equal parts  
If both parts contain 1-1 elements, compare them and put the min into the 1st slot one level back:  
  
  
Then go again:



Here we compared 541 first to 206, then to placed the min() into the 1st box  
Then compared it to 978 and put the min() into the 2nd box  
978 remained



Now we compare:  
- 206 and 385 (206 wins)  
- 541 and 385 (385 wins)  
- 541 and 695  
- 978 and 698  
- 978