Bubble Sort (Exchange)

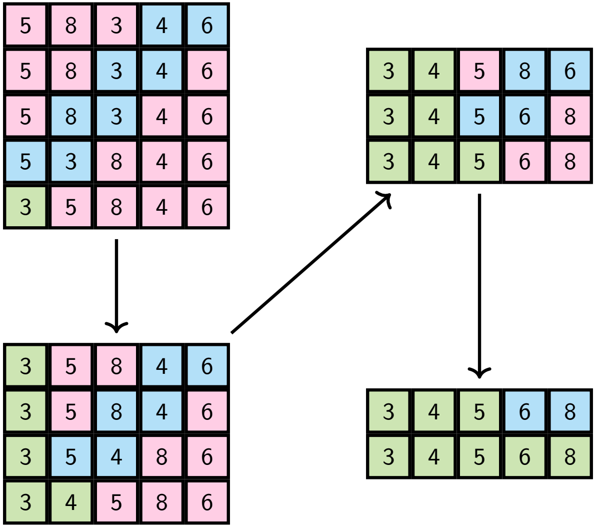
Bubble Sort

Bubble sort does multiple passes over an array of items, swapping neighbouring items that are out of order as it goes.

Each pass guarantees that at least one extra element ends up in its correct ordered location at the start of the array, so consecutive passes shorten to work only on the unsorted part of the array until the last pass only needs to sort the remaining two elements at the end of the array.

```
bubblesort(a, n) {
  for ( i = 1 ; i < n ; i++ )
     for ( j = n-1 ; j >= i ; j-- )
        if ( a[j] < a[j-1] )
           swap a[j] and a[j-1]
}
```

Example of a Bubble Sort run



Bubble Sort Complexity

The outer loop is iterated $n-1$ times.

The inner loop is iterated $n-i$ times, with each iteration executing a single comparison. So the total number of comparisons is:

$$\begin{aligned}\sum_{i=1}^{n-1}\sum_{j=i}^{n-1} 1 &= \sum_{i=1}^{n-1}(n-i) \\ &= (n-1)+(n-2)+\cdots+1 \\ &= \frac{n(n-1)}{2}.\end{aligned}$$

Thus best, average and worst case complexities are all $O(n^2)$

Bubble Sort Stability

Consider what happens when two elements with the same value are in the array to be sorted.

Since only neighbouring pairs of values can be swapped, and the swap is only carried out if one is strictly less than the other, no pair of the same values will ever be swapped. Hence bubble sort can not change the relative order of two elements with the same value.

Hence bubble sort is *stable*.