

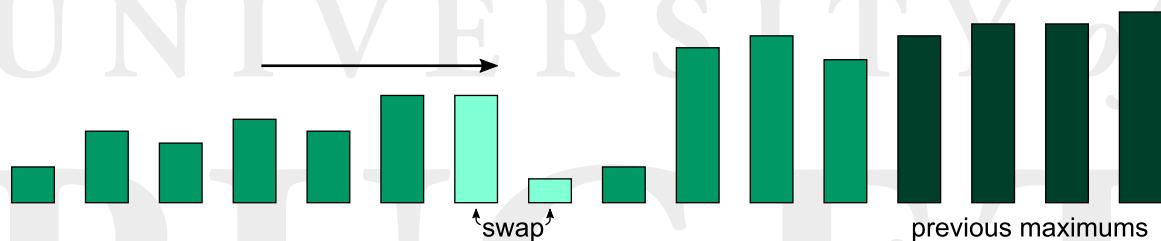
## Homework 6: Bubble Sort

Create an array of random numbers, and then sort them using bubble sort.

This assignment is very similar to the lab; the only difference is that you will be implementing bubble sort rather than insertion sort. As before you will make an int array of the size the user requests, where each element is between 1 and the size  $n$ . You will then sort it and report how many milliseconds it took, printing the array before and after sorting if its length is 100 or less.

```
Please enter the array size: 16
Unsorted: [14, 3, 7, 14, 9, 6, 9, 5, 13, 6, 15, 2, 3, 15, 16, 12]
Sorted: [2, 3, 3, 5, 6, 6, 7, 9, 9, 12, 13, 14, 14, 15, 15, 16]
Time taken to sort: 0 ms
```

Bubble sort works by comparing adjacent elements in the array, swapping them if they are out of order. It does this for every adjacent pair from the beginning to the end, so that the maximum element “floats” to the top one step at a time. Then it repeats the whole process, so that the second-greatest element floats up to the second-to-last place. It keeps doing this, so that each subsequent element will float up in turn.



Bubble sort will usually perform more swaps than selection sort. However, it can do better in cases where the array is already semi-sorted.

Here are some hints:

- Check all the hints from this week’s lab. Each of them is valid for this assignment as well. In fact, you’ll be able to reuse a lot of your code.
- Once you have it working, play with different array sizes. See if you can discover how the time taken varies by the size of the array. For example, when you double the array size, how does this affect the time taken to sort it?

This class will be called `BubbleSort`.

**Extra credit:** Usually, each pass in bubble sort goes in the same direction (low to high). However, if you alternate directions, you will be doing a different sort called shaker sort. Shaker sort is at least as fast as bubble sort, and is sometimes able to be significantly faster. If you implement this instead, you will get a modest amount of extra credit. If you do this, please rename your class `ShakerSort`.