

# Bubblesort

---

## 1 Algorithm

"Bubble" the largest value in the array to the top by a series of exchanges, then recurse (so to say) on the subarray  $A[:-1]$ .

```
1 for i in [n, 2]:  
    for j in [1, i-1]:  
3         if A[j] > A[j+1]:  
            A[j], A[j+1] = A[j+1], A[j]
```

## 2 Analysis

Bubblesort is  $O(n^2)$  in its best, worst, and average cases.

### 2.1 Comparisons

$$\sum_{i=2}^n \sum_{j=1}^{i-1} 1 = \sum_{i=2}^{n-1} i = \frac{n(n-1)}{2} \quad (1)$$

### 2.2 Exchanges

**Best case** The best case for bubblesort is a sorted array, which means 0 exchanges.

**Worst case** The worst case for bubble sort is a reverse-sorted array, which means every element goes through the maximum number of exchanges (equal to the number of comparisons).

$$\frac{n(n-1)}{2} \quad (2)$$

**Average case** Because the number of exchanges to conduct is equal to the number of transpositions in the array (reverse-sorted pairs) and all permutations are equally likely, picking a random permutation would expect to yield  $max/2$  transpositions.

$$\frac{n(n-1)}{4} \quad (3)$$