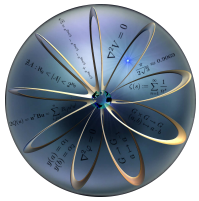

planetmath.org

Math for the people, by the people.

bubblesort

Canonical name	Bubblesort
Date of creation	2013-03-22 12:31:07
Last modified on	2013-03-22 12:31:07
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	9
Author	CWoo (3771)
Entry type	Algorithm
Classification	msc 68P10
Synonym	bubble sort
Related topic	SortingProblem
Related topic	StableSortingAlgorithm
Related topic	InPlaceSortingAlgorithm

The *bubblesort* algorithm is a simple and naïve approach to the sorting problem. Let $\leq$ define a total ordering over a list A of n values. The bubblesort consists of advancing through A, swapping adjacent values $A[i]$ and $A[i+1]$ if $A[i+1] \leq A[i]$ holds. By going through all of A in this manner n times, one is guaranteed to achieve the proper ordering.

Pseudocode

The following is pseudocode for the bubblesort algorithm. Note that it keeps track of whether or not any swaps occur during a traversal, so that it may terminate as soon as A is sorted.

```
BubbleSort(A, n, ≤)
Input: List A of n values.
Output: A sorted with respect to relation ≤.
Procedure:
done ← false
while (not done) do
      done ← true
      for i ← 0 to n − 1 do
          if A[i + 1] ≤ A[i]
            then swap(A[i], A[i + 1])
                 done ← false
          fi
      od
od
```

Analysis

The worst-case scenario is when A is given in reverse order. In this case, exactly one element can be put in order during each traversal, and thus all n traversals are required. Since each traversal consists of $n-1$ comparisons, the worst-case complexity of bubblesort is $\mathcal{O}(n^2)$.

Bubblesort is perhaps the simplest sorting algorithm to implement. Unfortunately, it is also the least efficient, even among $\mathcal{O}(n^2)$ algorithms. Bubblesort can be shown to be a stable sorting algorithm (since two items of equal keys are *never* swapped, initial relative ordering of items of equal keys is preserved), and it is clearly an in-place sorting algorithm.