



planetmath.org

Math for the people, by the people.

selection sort

Canonical name	SelectionSort
Date of creation	2013-03-22 11:44:42
Last modified on	2013-03-22 11:44:42
Owner	mathcam (2727)
Last modified by	mathcam (2727)
Numerical id	8
Author	mathcam (2727)
Entry type	Algorithm
Classification	msc 68P10
Classification	msc 05C50
Classification	msc 05C25
Classification	msc 83-01
Related topic	InsertionSort
Related topic	SortingProblem

## The Problem

See the Sorting Problem.

## The Algorithm

Suppose  $L = \{x_1, x_2, \dots, x_n\}$  is the initial list of unsorted elements. The *selection sort* algorithm sorts this list in  $n$  steps. At each step  $i$ , find the largest element  $L[j]$  such that  $j < n - i + 1$ , and swap it with the element at  $L[n - i + 1]$ . So, for the first step, find the largest value in the list and swap it with the last element in the list. For the second step, find the largest value in the list up to (but not including) the last element, and swap it with the next to last element. This is continued for  $n - 1$  steps. Thus the selection sort algorithm is a very simple, in-place sorting algorithm.

## Pseudocode

**Algorithm** SELECTION\_SORT( $L$ ,  $n$ )

*Input:* A list  $L$  of  $n$  elements

*Output:* The list  $L$  in sorted order

```
begin
  for  $i \leftarrow n$  downto 2 do
    begin
       $temp \leftarrow L[i]$ 
       $max \leftarrow 1$ 
      for  $j \leftarrow 2$  to  $i$  do
        if  $L[j] > L[max]$  then
           $max \leftarrow j$ 
       $L[i] \leftarrow L[max]$ 
       $L[max] \leftarrow temp$ 
    end
  end
```

## Analysis

The selection sort algorithm has the same runtime for any set of  $n$  elements, no matter what the values or order of those elements are. Finding the maximum element of a list of  $i$  elements requires  $i - 1$  comparisons. Thus  $T(n)$ ,

the number of comparisons required to sort a list of  $n$  elements with the selection sort, can be found:

$$\begin{aligned}
 T(n) &= \sum_{i=2}^n (i-1) \\
 &= \sum_{i=1}^n i - n - 2 \\
 &= \frac{(n^2 - n - 4)}{2} \\
 &= \mathcal{O}(n^2)
 \end{aligned}$$

However, the number of data movements is the number of swaps required, which is  $n - 1$ . This algorithm is very similar to the insertion sort algorithm. It requires fewer data movements, but requires more comparisons.