**Selection sort** is a simple sorting algorithm. This sorting algorithm is a in-place comparison based algorithm in which the list is divided into two parts, sorted part at left end and unsorted part at right end. Initially sorted part is empty and unsorted part is entire list.

Smallest element is selected from the unsorted array and swapped with the leftmost element and that element becomes part of sorted array. This process continues moving unsorted array boundary by one element to the right.

This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n2) where n is no. of items.

How Does it work?

Till the first position in the sorted list, the whole list is scanned sequentially.

JAVA Implementation:

public static void SelectionSort ( int [ ] num )  
{  
     int i, j, first, temp;    
     for ( i = num.length - 1; i > 0; i - - )    
     {  
          first = 0;  //initialize to subscript of first element  
          for(j = 1; j <= i; j ++)   //locate smallest element between positions 1 and i.  
          {  
               if( num[ j ] < num[ first ] )          
                 first = j;  
          }  
          temp = num[ first ];   //swap smallest found with element in position i.  
          num[ first ] = num[ i ];  
          num[ i ] = temp;   
      }             
}