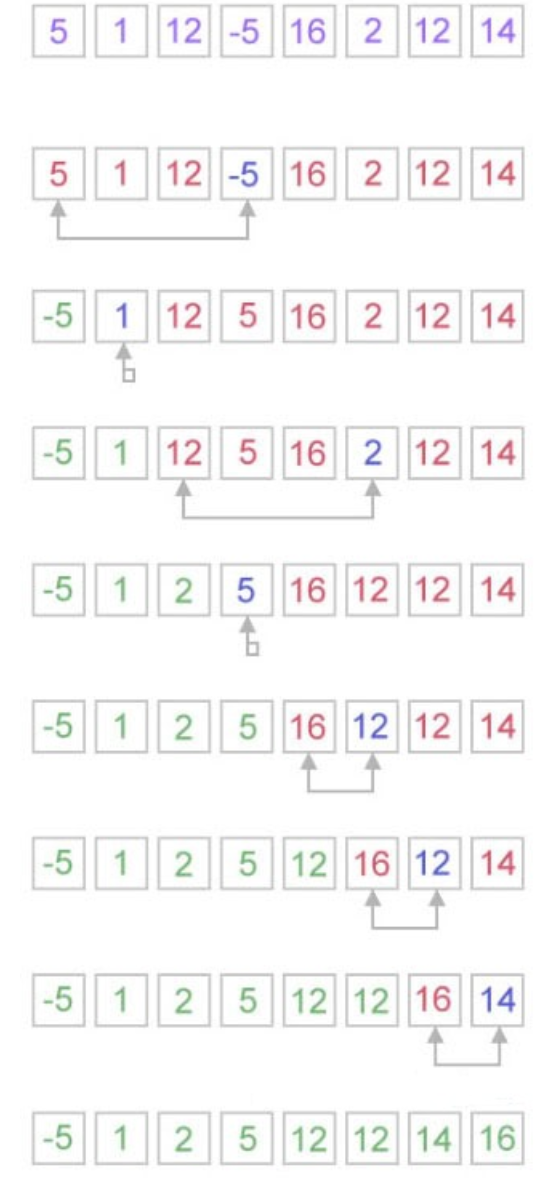
**Program: Implement selection sort in java.**

The selection sort is a combination of searching and sorting. During each pass, the unsorted element with the smallest (or largest) value is moved to its proper position in the array. The number of times the sort passes through the array is one less than the number of items in the array. In the selection sort, the inner loop finds the next smallest (or largest) value and the outer loop places that value into its proper location.



Selection sort is not difficult to analyze compared to other sorting algorithms since none of the loops depend on the data in the array. Selecting the lowest element requires scanning all n elements (this takesn − 1 comparisons) and then swapping it into the first position. Finding the next lowest element requires scanning the remaining n − 1 elements and so on, for (n − 1) + (n − 2) + ... + 2 + 1 = n(n − 1) / 2 ∈ Θ(n2) comparisons. Each of these scans requires one swap for n − 1 elements.

* ----

public static void selectionSort(int[] arr){

    /\*По очереди будем просматривать все подмножества

элементов массива (0 - последний, 1-последний,

2-последний,...)\*/

for (int i = 0; i < arr.length; i++) {

/\*Предполагаем, что первый элемент (в каждом

подмножестве элементов) является минимальным \*/

       int min = arr[i];

        int min\_i = i;

/\*В оставшейся части подмножества ищем элемент,

который меньше предположенного минимума\*/

        for (int j = i+1; j < arr.length; j++) {

//Если находим, запоминаем его индекс

            if (arr[j] < min) {

                min = arr[j];

                min\_i = j;

            }

       }

/\*Если нашелся элемент, меньший, чем на текущей позиции,

меняем их местами\*/

        if (i != min\_i) {

            int tmp = arr[i];

            arr[i] = arr[min\_i];

            arr[min\_i] = tmp;

        }

     }

}