**Bubble**

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

for (int j = 0 ; j < array.length – i – 1 ; j++) {

if (array[j] > array[j+1])

{

int temp = array[j];

array[j] = array[j+1];

array[j+1] = temp;

}

}

}

**QuickSort**

class QuickSort

{

    int partition(int arr[], int low, int high)

    {

        int pivot = arr[high];

        int i = (low-1); // index of smaller element

        for (int j=low; j<high; j++)

        {

            if (arr[j] <= pivot)

            {

                i++;

                // swap arr[i] and arr[j]

                int temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

        // swap arr[i+1] and arr[high] (or pivot)

        int temp = arr[i+1];

        arr[i+1] = arr[high];

        arr[high] = temp;

        return i+1;

    }

    void sort(int arr[], int low, int high)

    {

        if (low < high)

        {

            /\* pi is partitioning index, arr[pi] is

              now at right place \*/

            int pi = partition(arr, low, high);

            // Recursively sort elements before

            // partition and after partition

            sort(arr, low, pi-1);

            sort(arr, pi+1, high);

        }

    }

}