**Bubble Sort**

for(i=N-1; i>=0; i--){

for(j=0; j<i; j++){

**/\* compares adjacent numbers \*/**

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

**/\* swaps the number if condition satisfies \*/**

tmp = array[j];

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

array[j+1] = tmp;

}

}

}

**Selection Sort**

for(i=0; i<N-1; i++){

minIndex = i;

for(j=i+1; j<N; j++){

**/\* compares adjacent numbers \*/**

if(array[j] < array[minIndex]){

minIndex = j;

}

}

**/\* swaps the number if condition satisfies \*/**

tmp = array[minIndex];

array[minIndex] = array[i];

array[i] = tmp;

}

**Insertion Sort**

for (i=1; i<N; i++) {

tmp = array[i];

j=i;

while(j>0 && tmp<array[j-1]){

array[j] = array[j-1];

j--;

}

array[j]=tmp;

}

**Quick Sort**

function quicksort('array'){

if length('array') ≤ 1

return 'array' **// an array of zero or one elements is already sorted**

select and remove a pivot element 'pivot' from 'array' **// see 'Choice of pivot' below**

create empty lists 'less' and 'greater'

for ('x' in 'array'){

if ('x' ≤ 'pivot')

append 'x' to 'less'

else

append 'x' to 'greater'

}

return concatenate(quicksort('less'), list('pivot'), quicksort('greater')) **// two recursive calls**

}