7. Sorting Algorithms

What is sorting?
Arranging items in ascending or descending order is called as sorting.
There are different types of sorting techniques each of which is good for some cases such as nearly sorted, reversed, random, etc.

Selection Sort Algorithm:
Here we repeatedly find the next largest (or smallest) element in the array and move it to its final position in the sorted array.
Example: Sort the numbers 6, 7,72, 4, 32, 65, 9, 56 using selection sort.

	0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	
Pass0	6	7	72	4	32	65	9	56	Original
Pass1	4	7	72	6	32	65	9	56	
Pass2	4	6	72	7	32	65	9	56	
Pass3	4	6	7	72	32	65	9	56	
Pass4	4	6	7	9	32	65	72	56	
Pass5									

```
void swap(int *x,int *y)
{
   int t;
   t=(*x);
   *x=(*y);
   *y=t;
}
void selectionSort (int *a, int n)
{
   int i, j, m, t;
   for (i = 0; i < n-1; i++)
   {
      m = i;
      for (j = i+1; j < n; j++)
            if (a[j] < a[m]) m = j;
      swap(&a[i],&a[m]);
   }
}
void main ()
{
   clrscr();
   int a[] = {4, 65, 2, -31, 0, 99, 2, 83, 782, 1};
   int n = 10;
   displayArray(a,n);cout<<endl;
   selectionSort(a, n);
   displayArray(a,n);
}
```

Output:
4 65 2 -31 0 99 2 83 782 1
-31 0 1 2 2 4 65 83 99 782

Bubble Sort Algorithm:
Here we repeatedly move the largest element to the highest index position of the array.
Example: Sort the numbers 6, 7,72, 4, 32, 65, 9, 56 using bubble sort.

	0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	
Pass0	6	7	72	4	32	65	9	56	Original
Pass1	6	7	4	32	65	9	56	72	
Pass2	6	4	7	32	9	56	65	72	
Pass3	4	6	7	9	32	56	65	72	
Pass4	4	6	7	9	32	56	65	72	
Pass5	4	6	7	9	32	56	65	72	Sorted

Pseudo Code:

```
bubbleSort(a[],n) //Let 'a' be an array containing 'n' items
   max = n-2
   swapped = true
   while (max>0 AND swapped=true)
      swapped = false
      for j = 0 to max
```

```
        if (a[j] > a[j + 1])
           swap(&a[j],&a[j+1])
           swapped = true
        end if
      next j
      max=max-1
   end while
```

C++ Code:

```
void bubbleSort(int *a, int n)
{
   int j;
   int max = n-2;
   int swapped = 1;
   while (max>0 && swapped)
   {
      swapped = 0;
      for (j = 0; j <= max; j++)
      {
         if (a[j] > a[j + 1])
         {
            swap(&a[j],&a[j+1]);
            swapped = 1;
             }
      }
      max--;
   }
}
```

Efficiency of the Sort Algorithms (Best, Worst and Average Case Comparison):

Name	Best	Average	Worst
Quicksort	$n \log n$	$n \log n$	n^2
Merge sort	$n \log n$	$n \log n$	$n \log n$
In-place merge sort	—	—	$n \left(\log n\right)^2$
Heapsort	$n \log n$	$n \log n$	$n \log n$
Insertion sort	n	n^2	n^2
Introsort	$n \log n$	$n \log n$	$n \log n$
Selection sort	n^2	n^2	n^2
Timsort	n	$n \log n$	$n \log n$
Shell sort	n	$n(\log n)^2$ or $n^{3/2}$	Depends on gap sequence; best known is $n(\log n)^2$
Bubble sort	n	n^2	n^2
Binary tree sort	n	$n \log n$	$n \log n$