



# Bubble Sort



# Sorting



- Sorting is an operation that segregates items into groups according to specified criterion.
- Two types of sorting are:
  - Internal Sorting: technique in which data resides in the memory of the computer.
    - Egs of internal sorting: Bubble Sort, Insertion Sort, Quick Sort, Heap Sort, Radix Sort, Selection sort
  - External sorting: technique in which data resides in external or secondary storage devices such as floppy disk, hard disk etc.
    - Egs of external sorting: External merge sort



# Sorting

- It is technique by which we expect the list of elements to be arranged in some specific manner.
- Ascending order: elements are arranged from low to high value.
- Descending order: elements are arranged from high to low value.
  - Sorting Books in Library (Dewey system)
  - Sorting Individuals by Height (Feet and Inches)
  - Sorting Movies in Blockbuster (Alphabetical)
  - Sorting Numbers (Sequential)



# Sorting



- Time Complexity is defined to be the time the computer takes to run a program (or algorithm in our case).
- Space complexity is defined to be the amount of memory the computer needs to run a program.
  - Complexity in general, measures the algorithms efficiency in internal factors such as the time needed to run an algorithm.
- External Factors (not related to complexity):
  - Size of the input of the algorithm
  - Speed of the Computer
  - Quality of the Compiler



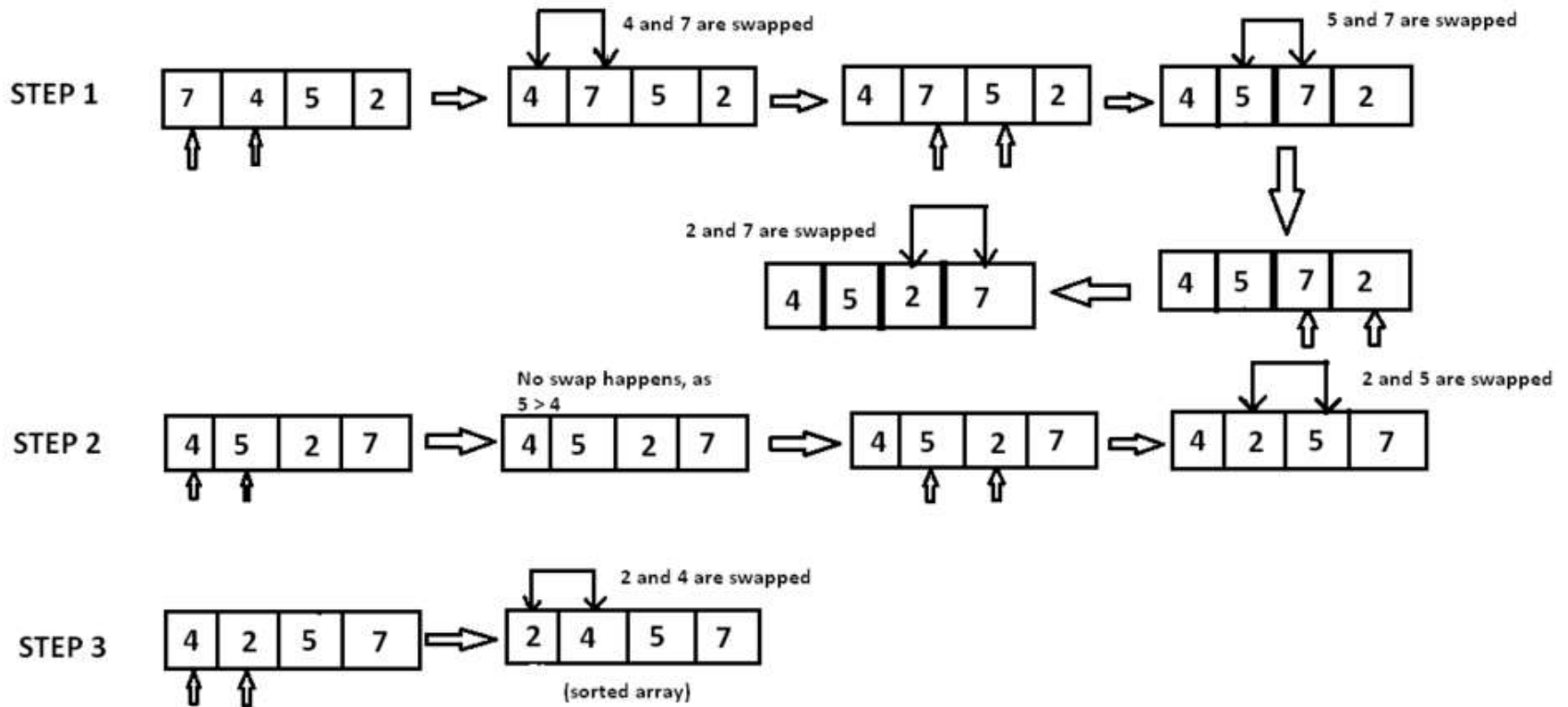
# Sorting: Time Complexity



Time complexity		Example
$O(1)$	<i>constant</i>	Adding to the front of a linked list
$O(\log N)$	<i>log</i>	Finding an entry in a sorted array
$O(N)$	<i>linear</i>	Finding an entry in an unsorted array
$O(N \log N)$	<i>n-log-n</i>	Sorting n items by 'divide-and-conquer'
$O(N^2)$	<i>quadratic</i>	Shortest path between two nodes in a graph
$O(N^3)$	<i>cubic</i>	Simultaneous linear equations



# Bubble Sort





# Bubble Sort Program

```
#include <stdio.h>
#include <conio.h>
int main()
{
int data[100],i,n,step,temp;
clrscr();
printf("Enter the number of elements to be sorted: ");
scanf("%d",&n);
for(i=0;i<n;++i)
{
printf("%d. Enter element: ",i+1);
scanf("%d",&data[i]);
}
}
```



# Bubble Sort Program

```
for(step=0;step<n-1;++step)
for(i=0;i<n-step-1;++i)
{
if(data[i]>data[i+1]) /* To sort in descending order, change > to < in this line. */
{
temp=data[i];
data[i]=data[i+1];
data[i+1]=temp;
}
}
printf("In ascending order: ");
for(i=0;i<n;++i)
printf("%d ",data[i]);
getch();
return 0;
}
```





# Bubble Sort Program

## **OUTPUT:**

Enter the number of elements to be sorted: 5

1. Enter element: 2

2. Enter element: 5

3. Enter element: 1

4. Enter element: 8

5. Enter element: 3

In ascending order: 1 2 3 5 8