


Check if array is sorted

4	8	13	16	20	25	28	33		
0	1	2	3	4	5	6	7	8	9

1. Inserting in a sorted array

- Suppose you want to insert 18 in the above sorted array, it must be done in such a way that all the values must be in sorted order even after inserting the new value
- The bits after 18 must be shifted accordingly

									
4	8	13	16	18	20	25	28	33	
0	1	2	3	4	5	6	7	8	9

Syntax for the above procedure is

Ex :

```
x = 18
i = length -1 ;
while( A[ i ] > x )
{
    A[ i + 1 ] = A[ i ]
    i - - ;
}
```

2. Checking if an array is sorted

- To check if all the elements in an array are sorted we can check if the present number is smaller than the next number if it is then it means that the list is sorted
- Algorithm for this is

EX:

```
Algorithm isSorted( A, n )
{
    for( i = 0 ; i < n -1 ; i ++ )
    {
        if(A[ i ] > A[ i+1 ])
            return false;
    }
    return true;
}
```

3. Arranging -ve on left side

- If we want to bring all -ve numbers on one side and all +ve numbers after that then for this we can take 2 pointers that is one for bringing all -ve at one place the other for moving +ve at one place
- The time taken for this is $O(n)$

Ex :

```
i = 0;
j = length - 1;

while( i < j )
{
    while ( A[ i ] < 0 ) { i++;}
    while ( A[ j ] >= 0 ) { j - - ;}
    if( i < j )
    {
        swap(A[ i ] , A[ j ]);
    }
}
```