Bubble Sort :

Bubble sort walks through the array, compares two elements at a time. If the elements are out of order, it swaps them. It continues to swap out of order elements until the entire collection is sorted.

Yes Bubble sort is in efficient algorithm, It takes Time Complexity as O(n^2) and do the n^2 swaps. It makes you algorithm runs slow.

There are two notes on Bubble Sort :

=> For every iteration the largest element would be swapped .
Assume currently nth iteration is running by this time n-1 elements would already been swapped.

=> If the input array is sorted then there won't be any swapping in bubble sort, this key we can use to bypass next iterations.

By considering above two points still the Bubble sort can be used on demand.

Bubble Sort :

i = 0 {7,4,5,2}
Inner loop
Sort j=0 ; j<arr.length;j++
=> {4,7,5,2}
=> {4,5,7,2}
=> {4,5,2,7}

i = 1 {4,5,2,7}
Inner loop
Sort j=0 ; j<arr.length ;j++
=> {4,5,2,7}
=> {4,2,5,7}
=> {4,2,5,7}

i = 2 {4,2,5,7}
Inner loop
Sort j=0 ; j<arr.length;j++
=> {2,4,5,7}
=> {2,4,5,7}
=> {2,4,5,7}

i = 3 {2,4,5,7}
Inner loop
Sort j=0 ; j<arr.length;j++
=> {2,4,5,7}
=> {2,4,5,7}
=> {2,4,5,7}

TimeComplexity :
Sorted Array => O(n)
Unsorted Array => O(n^2)
Space Complexity => O(1)
Stability => Stable
Comparison Sort =>Yes
Swap => O(n^2)
NonRecursive
Internal Sort

If the input array has all unsorted elements then Bubble sort is not recommended. It's not only takes time complexity O(n^2) even the swap happens n^2 times.