


✓ BUBBLE SORT ALGORITHM

Define = Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.

Advantages of Bubble Sort: 1. Bubble sort is easy to understand and implement. 2. It does not require any additional memory space. 3. It is a stable sorting algorithm, meaning that elements with the same key value maintain their relative order in the sorted output.

Disadvantages of Bubble Sort: 1. Bubble sort has a time complexity of $O(N^2)$ which makes it very slow for large data sets. 2. Bubble sort is a comparison-based sorting algorithm, which means that it requires a comparison operator to determine the relative order of elements in the input data set. It can limit the efficiency of the algorithm in certain cases.


```
import array
from array import *
def swap(array):
    n=len(array)
    for x in range(n):
        for y in range(n-x-1):
            if array[y]>array[y+1]:
                array[y],array[y+1]=array[y+1],array[y]
    print(array)
p=array('i',[5,4,3,2,1])
swap(p)
```

 array('i', [1, 2, 3, 4, 5])

✓ In descending order

```
import array
from array import *
b=array('i',[4,5,2,1,3])
l= len(b)
def bubble(a):
    for x in range(l):
        for y in range(l-x-1):
            if a[y]<a[y+1]:
                a[y],a[y+1]=a[y+1],a[y]
    print(a)
```

bubble(b)

 array('i', [5, 4, 3, 2, 1])