Insertion sort

Given an array of integers, sort the array using insertion sort Algorithm

0th index as a single number is always sorted.

build up the answer from there;

at each iteration move to the right and compare the current element with all its previous element, if smaller than previous swap them.

#Pseudo code

```
def sort(arr):
    n = len(arr)
    // one single element is always sorted, so consider 0th sorted
    for i in range(1, n)
        for j in range(i-1, -1, -1)
            // if prev numbers > current number
            if arr[j] > arr[j+1]:
                temp = arr[j+1]
                arr[j+1] = arr[j]
                arr[j] = temp
            else:
                break
```

```
def sort(arr):
    n = len(arr)
    for i in range(1, n):
        for j in range(i-1, -1, -1):
            if arr[j] > arr[j+1]:
                temp = arr[j+1]
                arr[j+1] = arr[j]
                arr[j] = temp
            else:
                break
```

i=1 j=(i-1,-1,-1) = (0,-1,-1)

arr= [9, 4, 1, 2, 3, -1]

arr[0] > arr[1], yes; swap them

arr= [4, 9, 1, 2, 3, -1]

i=2 j=(1,-1,-1)

arr[1] > arr[2]? yes; swap them

arr= [4, 1, 9, 2, 3, -1]

i = i-1 = 1-1=0

arr[0] > arr[1]? yes, swap them

i=3 j(2,-1,-1)

arr[2] > arr[3]? yes; swap them

arr= [1, 4, 2, 9, 3, -1]

j-=1 ⇒ 2

arr[1] > arr[2]? yes; swap them

arr= [1, 2, 4, 9, 3, -1]

j-=1 ⇒ 1

arr[0] > arr[1]? No, break

i=4 ; j=3

arr= [1, 2, 4, 9, 3, -1]

arr[3] > arr[4]? yes, swap them

arr= [1, 2, 4, 3, 9, -1]

j-=1 ⇒3

arr[2] > arr[3]? yes; swap them

arr= [1, 2, 3, 4, 9, -1]

j-=1 ⇒ 2

arr[1] > arr[2]? No; break

i=5 j=4

arr= [1, 2, 3, 4, 9, -1]
(0 1 2 3 4 5)

arr[4] > arr[3]? yes; swap them

arr= [1, 2, 3, 4, -1, 9]

j-=1 ⇒ 3

arr[3] > arr[4]? yes; swap them

arr= [1, 2, 3, -1, 4, 9]

j-=1 ⇒ 2

arr[2] > arr[3]? yes; swap them

arr= [1, 2, -1, 3, 4, 9]

j-=1 = 1

arr[1] > arr[2]? yes; swap them

arr= [1, -1, 2, 3, 4, 9]

i-=1 ⇒ 0

arr[0] > arr[1]? yes; swap them

arr= [-1, 1, 2, 3, 4, 9]