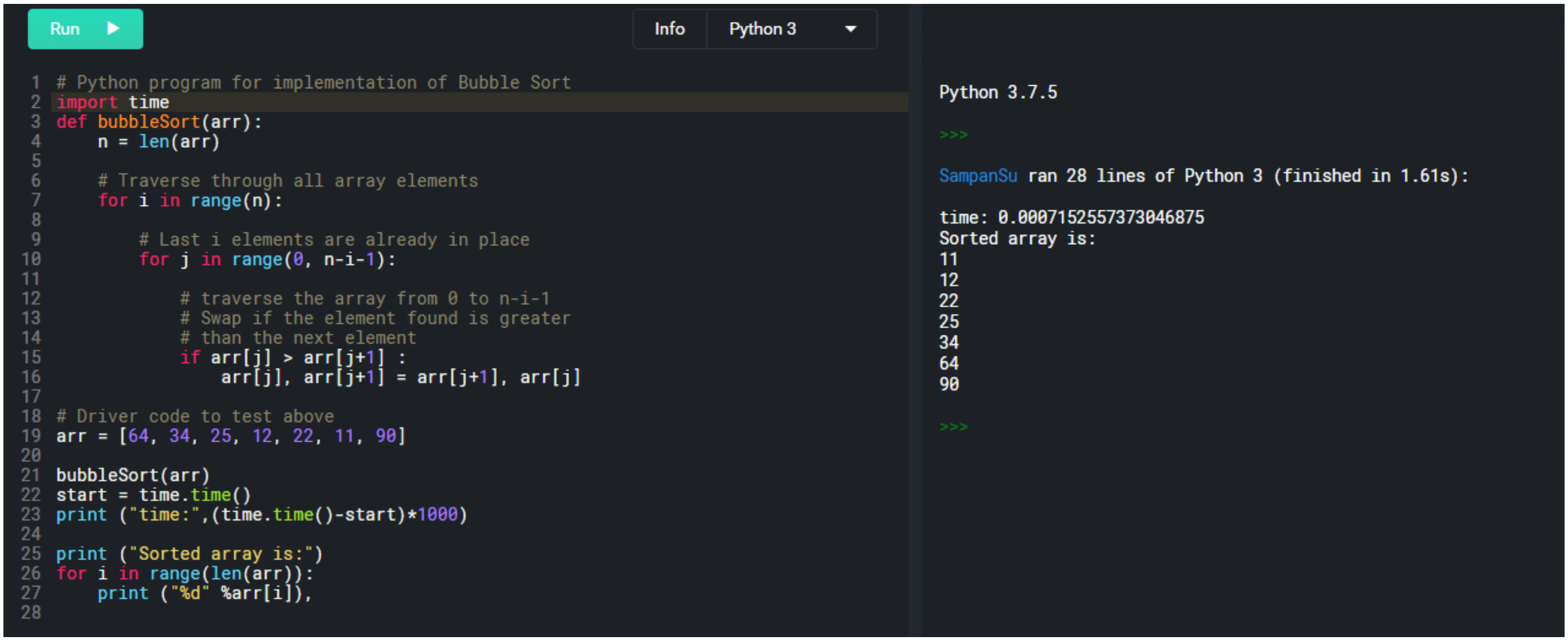
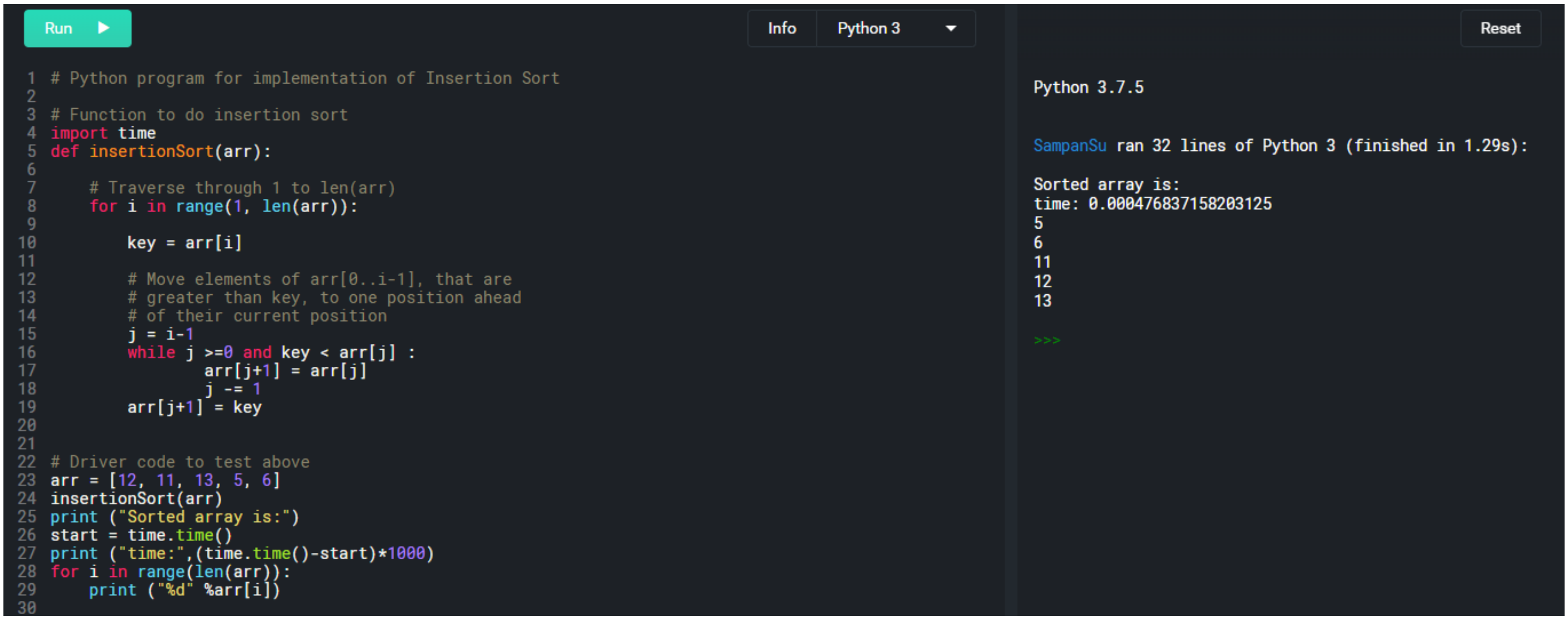
Buble sort



```python
# Python program for implementation of Bubble Sort
import time
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):

        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)
start = time.time()
print ("time:",(time.time()-start)*1000)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i]),

```

```
Python 3.7.5

>>>

SampanSu ran 28 lines of Python 3 (finished in 1.61s):

time: 0.0007152557373046875
Sorted array is:
11
12
22
25
34
64
90

>>>
```

Insertion sort



```python
# Python program for implementation of Insertion Sort

# Function to do insertion sort
import time
def insertionSort(arr):

    # Traverse through 1 to len(arr)
    for i in range(1, len(arr)):

        key = arr[i]

        # Move elements of arr[0..i-1], that are
        # greater than key, to one position ahead
        # of their current position
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
        arr[j+1] = key


# Driver code to test above
arr = [12, 11, 13, 5, 6]
insertionSort(arr)
print ("Sorted array is:")
start = time.time()
print ("time:",(time.time()-start)*1000)
for i in range(len(arr)):
    print ("%d" %arr[i])

```

```
Python 3.7.5

SampanSu ran 32 lines of Python 3 (finished in 1.29s):

Sorted array is:
time: 0.000476837158203125
5
6
11
12
13

>>>
```