

Ταξινομηση

# Ταξινόμηση με εισαγωγή

## Insertion Sort

- Διαφορετική μέθοδος ταξινόμησης από την ταξινόμηση με εισαγωγή
- Σε κάθε βήμα της ταξινόμησης χωρίζουμε και πάλι τη λίστα σε δύο μέρη ένα **ταξινομημένο** και ένα **μη ταξινομημένο**
- Ολοκληρώνουμε όταν το μέγεθος του μη ταξινομημένου μηδενιστεί.

# Ταξινόμηση με εισαγωγή

## Insertion Sort

- Σε κάθε βήμα της ταξινόμησης χωρίζουμε τη λίστα σε δύο μέρη ένα **ταξινομημένο** και ένα **μη ταξινομημένο**

1	2	5	10	29	25	12	21	15
---	---	---	----	----	----	----	----	----

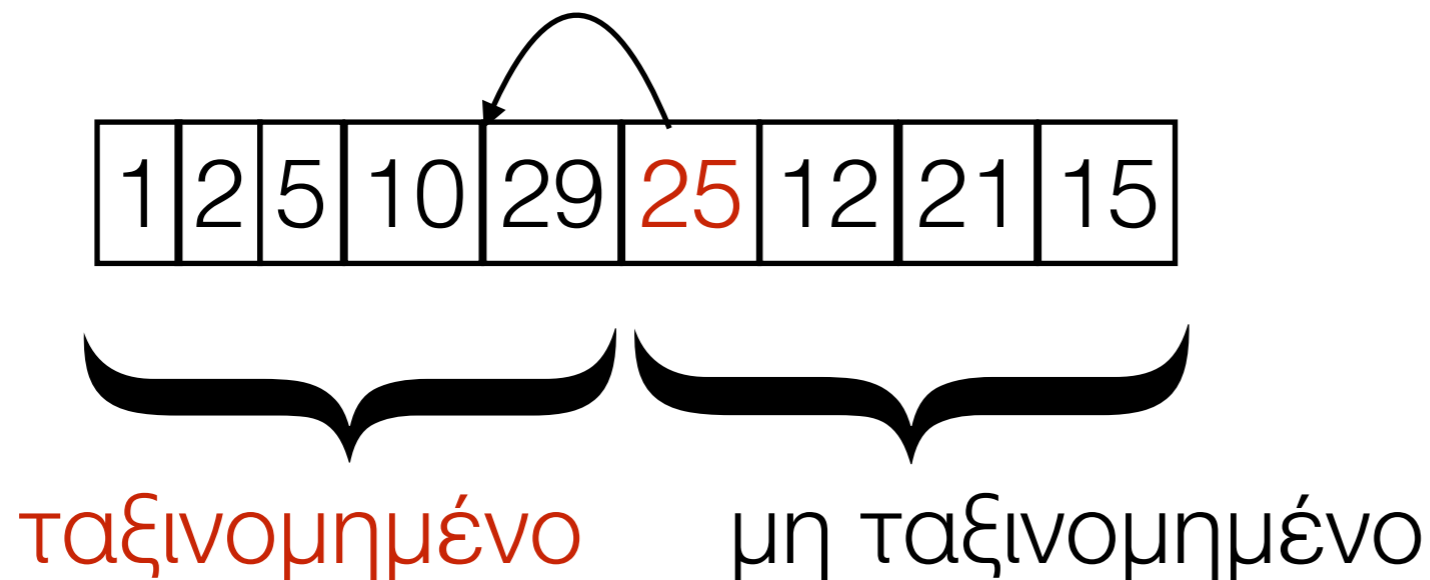


**ταξινομημένο**

μη ταξινομημένο

# Ταξινόμηση με εισαγωγή

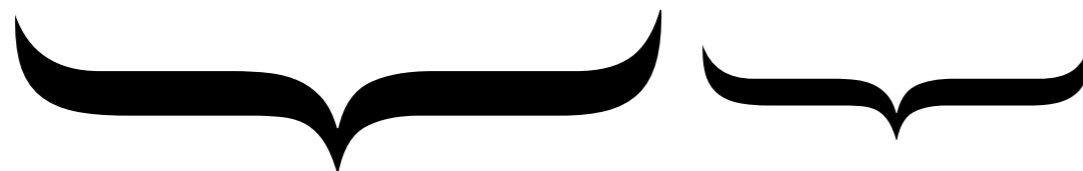
- Τοποθετούμε το πρώτο στοιχείο του μη ταξινομημένου μέρους στην κατάλληλη θέση του πρώτου τμήματος



# Ταξινόμηση με εισαγωγή

- Το ταξινομημένο μέρος αυξάνετε και ελαττώνετε αντιστοίχα το μη ταξινομημένο

1	2	5	10	25	29	12	21	15
---	---	---	----	----	----	----	----	----

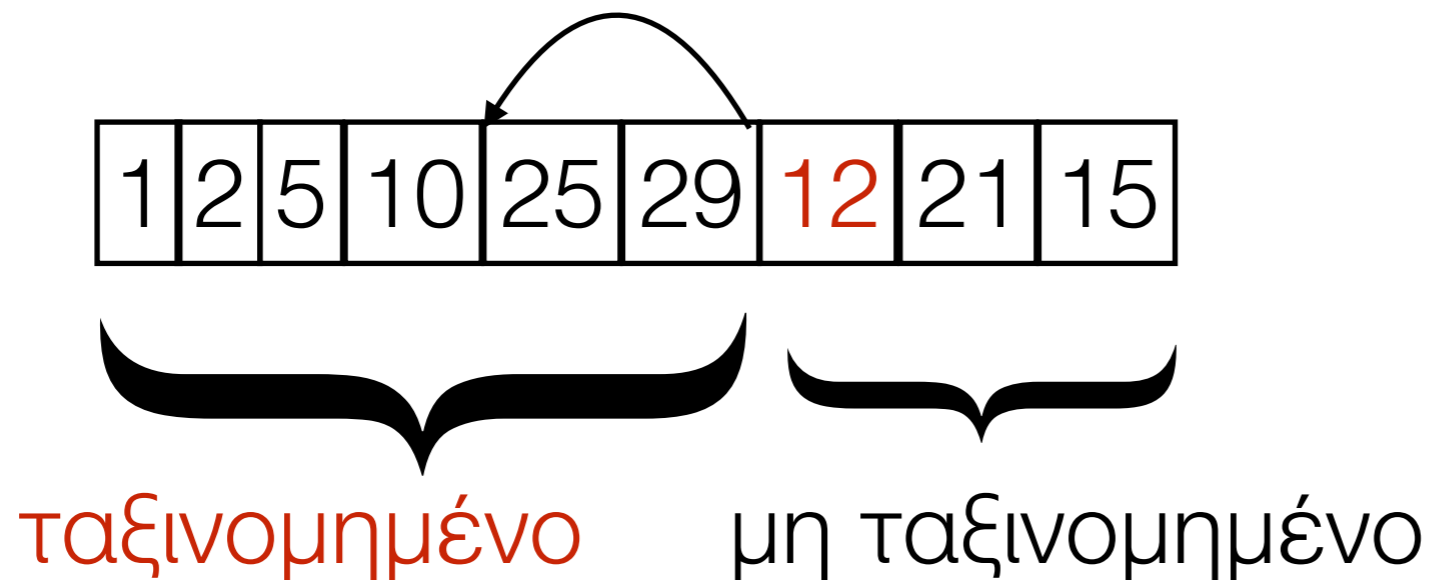


ταξινομημένο

μη ταξινομημένο

# Ταξινόμηση με εισαγωγή

- Συνεχίζουμε με το επόμενο πρώτο στοιχείο του μη ταξινομημένου μέρους



# Ταξινόμηση με εισαγωγή

```
def insertion_sort(L):  
    """Reorder the values in L from smallest to  
    largest."""  
  
    i = 0 ← μετρητής. Πόσα στοιχεία έχω ταξινομήσει  
    while i != len(L):  
        # Βάλτε το στοιχείο L[i] στην κατάλληλη θέση στο κομμάτι της λίστας L[0:i+1].  
        i = i + 1 ← Αυξάνω το μετρητή
```

# Ταξινόμηση με εισαγωγή

```
def insertion_sort(L):  
    """Reorder the values in L from smallest to  
    largest."""  
  
    i = 0  
    while i != len(L):  
        insert(L, i) ←  
        i = i + 1
```

συνάρτηση που τοποθετεί το  
στοιχείο της θέσης  $i$  στην  
κατάλληλη ταξινομημένη θέση  
στο κομμάτι από τη θέση 0 έως  $i$

```
def insert(L, b):  
    """Put L[b] where it belongs in L[0:b + 1].  
    Assume L[0:b] is sorted."""
```



# Ταξινόμηση με εισαγωγή

```
def insertion_sort(L):  
    """Reorder the values in L from smallest to largest."""  
  
    i = 0  
    while i != len(L):  
        insert(L, i)  
        i = i + 1  
  
def insert(L, b):  
    """Put L[b] where it belongs in L[0:b + 1].  
    Assume L[0:b] is sorted."""  
  
    # Ψάχνω τα στοιχεία από τη θέση b προς τα πίσω για ένα στοιχείο μικρότερο από το L[b].  
    i = b  
    while i != 0 and L[i - 1] >= L[b]:  
        i = i - 1
```

# Ταξινόμηση με εισαγωγή

```
def insertion_sort(L):  
    """Reorder the values in L from smallest to largest."""  
  
    i = 0  
    while i != len(L):  
        insert(L, i)  
        i = i + 1  
  
def insert(L, b):  
    """Put L[b] where it belongs in L[0:b + 1].  
    Assume L[0:b] is sorted."""  
  
    i = b  
    while i != 0 and L[i - 1] >= L[b]:  
        i = i - 1  
  
    # Μετακινώ το L[b] στη θέση με δείκτη i, μεταφέρω όλες τις τιμές προς τα “δεξιά”
```

# Ταξινόμηση με εισαγωγή

```
def insertion_sort(L):
    """Reorder the values in L from smallest to largest."""

    i = 0
    while i != len(L):
        insert(L, i)
        i = i + 1

def insert(L, b):
    """Put L[b] where it belongs in L[0:b + 1].
    Assume L[0:b] is sorted."""

    i = b
    while i != 0 and L[i - 1] >= L[b]:
        i = i - 1

    # Μετακινώ το L[b] στη θέση με δείκτη i, μεταφέρω όλες τις τιμές προς τα "δεξιά"
    value = L[b]
    del L[b]
    L.insert(i, value)
```