

ΠΡΟΤΕΙΝΟΜΕΝΕΣ  
ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ

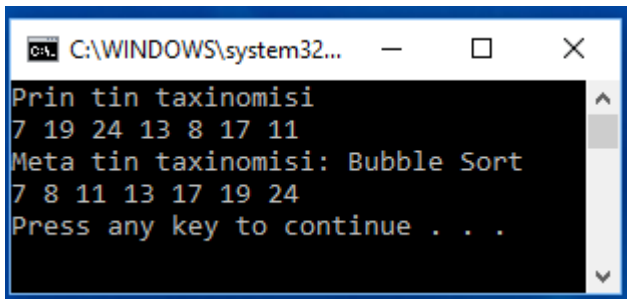
Αλγόριθμοι Ταξινόμησης

**ΑΣΚΗΣΗ-1<sup>η</sup>**

**Bubble Sort**

```
class BubbleSort {
    public static void main(String[] args) {
        int nums[] = {7,19,24,13,8,17,11};
        System.out.println("Printin taxinomisi");
        for (int i = 0; i < nums.length; i++)
            System.out.print(nums[i]+" ");
        System.out.println();
        BSort(nums);
        System.out.println("Metat in taxinomisi");
        for (int i = 0; i < nums.length; i++)
            System.out.print(nums[i]+" ");
        System.out.println(); }

    public static void BSort(int[] A) {
        int i, j;
        int temp;
        boolean flag;
        for (i=1; i < A.length; i++) {
            flag=true;
            for (j=0; j < A.length-i; j++)
                if (A[j] > A[j+1]) {
                    temp=A[j];
                    A[j]= A[j+1];
                    A[j+1]=temp;
                    flag=false; }
            if (flag) return; }}}
```



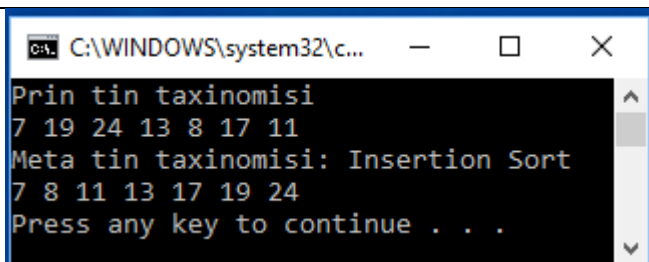
```
C:\WINDOWS\system32... - □ ×
Prin tin taxinomisi
7 19 24 13 8 17 11
Meta tin taxinomisi: Bubble Sort
7 8 11 13 17 19 24
Press any key to continue . . .
```

## ΑΣΚΗΣΗ-2<sup>η</sup>

### Insertion Sort

```
class InsertionSort {
public static void main(String[] args) {
    int nums[] = {7,19,24,13,8,17,11};
    System.out.println("Prin tin taxinomisi");
    for (int i = 0; i < nums.length; i++)
        System.out.print(nums[i]+" ");
    System.out.println();
InsSort(nums);
    System.out.println("Meta tin taxinomisi: Insertion Sort");
    for (int i = 0; i < nums.length; i++)
        System.out.print(nums[i]+" ");
    System.out.println(); }

public static void InsSort(int[] A) {
    for (int i=1; i < A.length; i++) {
        int current = A[i];
        int j=i;
        // metakinisi megalyteron timon dexia
        while (j>0 && A[j-1] > current) {
            A[j] = A[j-1];
            j--;}
        A[j] = current; }
}}}
```



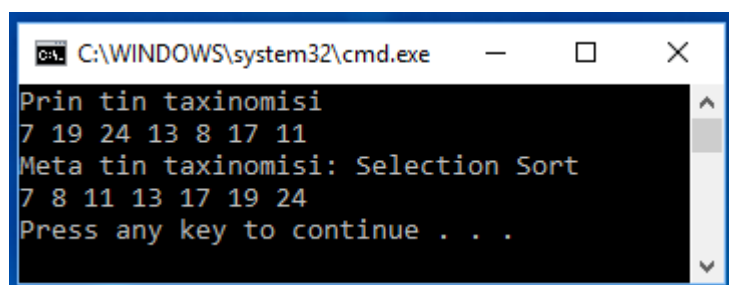
```
C:\WINDOWS\system32\c... - □ ×
Prin tin taxinomisi
7 19 24 13 8 17 11
Meta tin taxinomisi: Insertion Sort
7 8 11 13 17 19 24
Press any key to continue . . .
```

## ΑΣΚΗΣΗ-3<sup>η</sup>

### Selection Sort

```
class SelectionSort {
    public static void main(String[] args) {
        int nums[] = {7,19,24,13,8,17,11};
        System.out.println("Prin tin taxinomisi");
        for (int i = 0; i < nums.length; i++)
            System.out.print(nums[i]+" ");
        System.out.println();
        SelSort(nums);
        System.out.println("Meta tin taxinomisi: Selection Sort");
        for (int i = 0; i < nums.length; i++)
            System.out.print(nums[i]+" ");
        System.out.println(); }

    public static void SelSort(int[] A) {
        int pos, temp;
        for (int i = 0; i < A.length-1; i++){
            // thesi tis min timis
            pos=i;
            for (int k=i+1; k<A.length; k++)
                if (A[k] < A[pos]) pos=k;
            //swap tis times "pos" kai "i"
            temp = A[pos];
            A[pos] = A[i];
            A[i] = temp; }
    } }
```



```
C:\WINDOWS\system32\cmd.exe
Prin tin taxinomisi
7 19 24 13 8 17 11
Meta tin taxinomisi: Selection Sort
7 8 11 13 17 19 24
Press any key to continue . . .
```

## ΑΣΚΗΣΗ-4<sup>η</sup>

### Αναδρομική γρήγορη ταξινόμηση (Quick Sort)

```
public class QuickSortLiang {
    public static void quickSort(int[] list) {
        quickSort(list, 0, list.length - 1); }

    public static void quickSort(int[] list, int first, int last) {
        if (last > first) {
            int pivotIndex = partition(list, first, last);
            quickSort(list, first, pivotIndex - 1);
            quickSort(list, pivotIndex + 1, last); } }

    /** Partition the array list[first..last] */
    public static int partition(int[] list, int first, int last) {
        int pivot = list[first];    //Choose the first element as the pivot
        int low = first + 1;        //Index for forward search
        int high = last;           //Index for backward search
        while (high > low) {
            // Search forward from left
            while (low <= high && list[low] <= pivot)
                low++; //ayxisi toy aristerou deikti
            // Search backward from right
            while (low <= high && list[high] > pivot)
                high--; //meiosi toy dexiou deikti
            // Swap two elements in the list
            if (high > low) {
                int temp = list[high];
                list[high] = list[low];
                list[low] = temp;
            } }

        while (high > first && list[high] >= pivot)
            high--;
        // Swap pivot with list[high]
        if (pivot > list[high]) {
            list[first] = list[high];
            list[high] = pivot;
            return high;}
        else {return first;}}
```

```

public static void main(String[] args) {
    int[] list = {5,2,9,3,8,4,0,1,6,7};
    quickSort(list);
    for (int i = 0; i < list.length; i++)
        System.out.print(list[i] + " ");
    System.out.println();
}}

```

```

0 1 2 3 4 5 6 7 8 9
Press any key to continue . . .

```

[Παραλλαγή αναδρομικής γρήγορης αναζήτησης \(χρήση αντικειμένου\)](#)

```

import java.util.Arrays;
public class QuickSortDemo{
    public static void main(String args[]) {
        int[] nums = {6, 5, 3, 1, 8, 7, 2, 4};
        System.out.println("Ataxinomitos pinakas");
        System.out.println(Arrays.toString(nums));
        System.out.println();
        QuickSort algorithm = new QuickSort();
        // taxinomisi toy pinaka me ton algorithmo quicksort
        algorithm.sort(nums);
        // emfanisi toy taxinomimenoy pinaka
        System.out.println();
        System.out.println("Taxinomimenos pinakas me Quick Sort ");
        System.out.println(Arrays.toString(nums));
        System.out.println(); }}}

class QuickSort {
    private int input[];
    private int length;
    public void sort(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            return; }

```

```
this.input = numbers;
length = numbers.length;
quickSort(0, length - 1); }
```

```
private void quickSort(int low, int high) {
```

```
int i = low;
```

```
int j = high;
```

```
int x=0;
```

```
x=low + (high - low) / 2;
```

```
int pivot = input[low + (high - low) / 2]; //pivot sti mesi
```

```
// Diairesi se 2 ypo-pinakes - partitioning
```

```
while (i <= j) {
```

```
/* Se kathe epanalipsi tha vriskoume apo ta aristera ena arithmo pou
```

```
* tha einai megalyteros toy pivot kai ena arithmo apo ta dexia pou
```

```
* tha einai mikroteros toy pivot. Otan teleiosei o elehos tha
```

```
* ginetai h antimetathesi(swap) ton timon */
```

```
//prohoroyme apo ta aristera pros dexia mexri na xreiastei ena swap
```

```
while (input[i] < pivot) {i++;}
```

```
//prohoroyme apo ta dexia pros ta aristera mexri na xreiastei ena swap
```

```
while (input[j] > pivot) {j--;}
```

```
if (i <= j) {
```

```
//kane to swap
```

```
swap(i, j);
```

```
// metakinise tous deiktes stin epomeni thesi kai stis 2 kateythinseis
```

```
i++;
```

```
j--; } }
```

```
// anadromiki klisi tis quickSort()
```

```
if (low < j) {quickSort(low, j); }
```

```
if (i < high) {quickSort(i, high);}
```

```
}
```

```
// H methodos antikatastasis
```

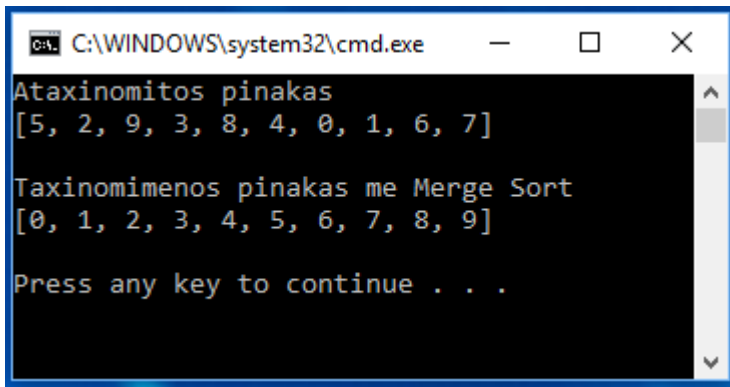
```
private void swap(int i, int j) {
```

```
int temp = input[i];
```

```
input[i] = input[j];
```

```
input[j] = temp;
```

```
}} }
```



```
C:\WINDOWS\system32\cmd.exe
Ataxinomitos pinakas
[5, 2, 9, 3, 8, 4, 0, 1, 6, 7]

Taxinomimenos pinakas me Merge Sort
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Press any key to continue . . .
```

## ΑΣΚΗΣΗ-5<sup>η</sup>

```
import java.util.Arrays;
public class MergeSort {
    /** H methodos taxinomei tous arithmous tou pinaka A */
    public static void mergeSort(int[] A) {
        if (A.length > 1) {
            // anadromiki Merge sort tou 1ou misou tou pinaka(copy)
            int[] firstHalf = new int[A.length/2]; //dhmioyrgia toy 1ou temp array
            System.arraycopy(A, 0, firstHalf, 0, A.length/2);
            mergeSort(firstHalf);
            // anadromiki Merge sort tou 2ou misou tou pinaka(copy)
            int secondHalfLength = A.length - A.length/2; //2o temp array
            int[] secondHalf = new int[secondHalfLength];
            System.arraycopy(A, A.length/2, secondHalf, 0, secondHalfLength);
            mergeSort(secondHalf);
            //Merge ton taxinomimenon 1ou-kai 2ou pinaka se ena temp-array
            merge(firstHalf, secondHalf, A);
        }
        /* Merge tous 2 taxinomimenous pinakes list1 kai list2
        * H methodos epanalptika sygkrinei ta stoixeia ton list1 kai list2
        * kai topothetei to mikrotero ston temp.
        public static void merge(int[] list1, int[] list2, int[] temp) {
            int current1 = 0; //trehon deiktis sto list1
            int current2 = 0; //trehon deiktis sto list2
            int current3 = 0; //trehon deiktis sto temp
            while (current1 < list1.length && current2 < list2.length) {
                if (list1[current1] < list2[current2])
                    /*an to mikrotero stoixeio vrethike sto list1 ayxise ton
                    * current1 kata 1, diaforetika ayxise ton current2 kata 1 */
                    temp[current3++] = list1[current1++];
```

```

else
    temp[current3++] = list2[current2++];
}
/* Ola ta stoixeia metaferontai ston temp
 * akomi kai ayta poy den metakinithikan
 * apo ta list1 kai list2 */
while (current1 < list1.length)
    temp[current3++] = list1[current1++];
    while (current2 < list2.length)
        temp[current3++] = list2[current2++];
}

```

```

public static void main(String[] args) {
    int[] nums = {5,2,9,3,8,4,0,1,6,7};
    System.out.println("Ataxinomitos pinakas");
    System.out.println(Arrays.toString(nums));
    System.out.println();
    mergeSort(nums);
    System.out.println("Taxinomimenos pinakas me Merge Sort ");
    System.out.println(Arrays.toString(nums));
    System.out.println(); } }

```

```

C:\WINDOWS\system32\cmd.exe
Ataxinomitos pinakas
[5, 2, 9, 3, 8, 4, 0, 1, 6, 7]

Taxinomimenos pinakas me Merge Sort
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Press any key to continue . . .

```