

Άσκηση 1: Απλή Ταξινόμηση Αριθμών

Περιγραφή: Δημιούργησε μια εφαρμογή C# που χρησιμοποιεί τον αλγόριθμο Bucket Sort για να ταξινομήσει έναν πίνακα αριθμών με δεκαδικές τιμές μεταξύ 0 και 1.

Κώδικας:

```
using System;
using System.Collections.Generic;

class BucketSortExercise1
{
    static void Main()
    {
        double[] numbers = { 0.78, 0.17, 0.39, 0.26, 0.72 };
        BucketSort(numbers);

        Console.WriteLine("Ταξινομημένοι αριθμοί:");
        foreach (var num in numbers)
        {
            Console.Write(num + " ");
        }
    }

    static void BucketSort(double[] array)
    {
        int n = array.Length;
        if (n <= 0)
            return;

        // Δημιουργούμε τα κουβάδες
        List<double>[] buckets = new List<double>[n];
        for (int i = 0; i < n; i++)
        {
            buckets[i] = new List<double>();
        }

        // Τοποθετούμε τους αριθμούς στους κουβάδες
        foreach (var num in array)
        {
            int index = (int)(num * n); // Υπολογίζουμε σε
            ποιον κουβά να βάλουμε τον αριθμό
        }
    }
}
```

```

        buckets[index].Add(num);
    }

    // Ταξινομούμε τους κουβάδες και τους
    συγκεντρώνουμε
    int pos = 0;
    foreach (var bucket in buckets)
    {
        bucket.Sort(); // Ταξινόμηση των αριθμών μέσα
        στον κουβά
        foreach (var num in bucket)
        {
            array[pos++] = num;
        }
    }
}

```

Λύση: Ο κώδικας δημιουργεί 5 κουβάδες, βάζει τους αριθμούς σε αυτούς, τους ταξινομεί και μετά τους συνδυάζει για να έχεις την τελική ταξινομημένη λίστα.

Άσκηση 2: Ταξινόμηση Μεγάλων Αριθμών

Περιγραφή: Δημιούργησε μια εφαρμογή C# που χρησιμοποιεί τον αλγόριθμο Bucket Sort για να ταξινομήσει έναν πίνακα από ακέραιους αριθμούς σε εύρος από 0 έως 100.

Κώδικας:

```

using System;
using System.Collections.Generic;

class BucketSortExercise2
{
    static void Main()
    {
        int[] numbers = { 42, 89, 25, 76, 55, 30, 12, 80 };
        BucketSort(numbers);
    }
}

```

```

Console.WriteLine("Ταξινομημένοι αριθμοί:");
foreach (var num in numbers)
{
    Console.Write(num + " ");
}
}

```

```

static void BucketSort(int[] array)
{

```

```

    int max = 100; // Το μέγιστο εύρος τιμών
    int n = array.Length;

```

```

    // Δημιουργούμε τα κουβάδες

```

```

    List<int>[] buckets = new List<int>[max + 1];

```

```

    for (int i = 0; i <= max; i++)
    {

```

```

        buckets[i] = new List<int>();
    }

```

```

    // Τοποθετούμε τους αριθμούς στους κουβάδες

```

```

    foreach (var num in array)
    {

```

```

        buckets[num].Add(num);
    }

```

```

    // Συγκεντρώνουμε τους αριθμούς από τους
    κουβάδες

```

```

    int pos = 0;

```

```

    foreach (var bucket in buckets)
    {

```

```

        foreach (var num in bucket)
        {

```

```

            array[pos++] = num;
        }
    }

```

```

}

```

Λύση: Ο κώδικας δημιουργεί 101 κουβάδες (από 0 έως 100), τοποθετεί τους αριθμούς στους αντίστοιχους

κουβάδες με βάση την τιμή τους, και στη συνέχεια συγκεντρώνει τους αριθμούς για να έχεις την ταξινομημένη λίστα.

Άσκηση 3: Ταξινόμηση Με Διαφορετικούς Κουβάδες

Περιγραφή: Δημιούργησε μια εφαρμογή C# που χρησιμοποιεί τον αλγόριθμο Bucket Sort για να ταξινομήσει έναν πίνακα από αριθμούς που κυμαίνονται από 0 έως 500 με 10 κουβάδες.

Κώδικας:

```
using System;
using System.Collections.Generic;

class BucketSortExercise3
{
    static void Main()
    {
        int[] numbers = { 300, 125, 475, 50, 200, 400, 100 };
        BucketSort(numbers, 10);

        Console.WriteLine("Ταξινομημένοι αριθμοί:");
        foreach (var num in numbers)
        {
            Console.Write(num + " ");
        }
    }
}
```

```

}

static void BucketSort(int[] array, int numBuckets)
{
    int max = 500; // Το μέγιστο εύρος τιμών
    int bucketSize = max / numBuckets;
    List<int>[] buckets = new List<int>[numBuckets];
    for (int i = 0; i < numBuckets; i++)
    {
        buckets[i] = new List<int>();
    }

    // Τοποθετούμε τους αριθμούς στους κουβάδες
    foreach (var num in array)
    {
        int index = num / bucketSize; // Υπολογίζουμε σε
        ποιον κουβά να βάλουμε τον αριθμό
        buckets[index].Add(num);
    }

    // Ταξινομούμε τους κουβάδες και τους
    συγκεντρώνουμε
    int pos = 0;
    foreach (var bucket in buckets)
    {
        bucket.Sort(); // Ταξινόμηση των αριθμών μέσα
        στον κουβά
        foreach (var num in bucket)
        {
            array[pos++] = num;
        }
    }
}

```

Λύση: Ο κώδικας δημιουργεί 10 κουβάδες, κάθε κουβάς καλύπτει ένα εύρος αριθμών, τοποθετεί τους αριθμούς στους κουβάδες με βάση την τιμή τους, τους ταξινομεί και μετά τους συνδυάζει για να έχεις την τελική ταξινομημένη λίστα.