

Άσκηση 1: Αναζήτηση με Αποθήκευση Θέσης

Περιγραφή: Φτιάξε μια κλάση που να περιέχει μια μέθοδο για βελτιστοποιημένη γραμμική αναζήτηση σε έναν πίνακα. Η μέθοδος πρέπει να αποθηκεύει την τελευταία θέση που βρέθηκε το στοιχείο ώστε να ξεκινά από εκεί την επόμενη φορά.

Κώδικας:

```
using System;

public class OptimizedLinearSearch
{
    private static int lastFoundIndex = -1;

    public static int Search(int[] array, int target)
    {
        int startIndex = lastFoundIndex + 1;

        for (int i = startIndex; i < array.Length; i++)
        {
            if (array[i] == target)
            {
                lastFoundIndex = i;
                return i;
            }
        }

        return -1; // Το στοιχείο δεν βρέθηκε
    }

    public static void Main()
    {
        int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        int target = 5;

        Console.WriteLine($"Αναζήτηση 1: Το στοιχείο {target} βρέθηκε στη θέση {Search(numbers, target)}.");
        Console.WriteLine($"Αναζήτηση 2: Το στοιχείο {target} βρέθηκε στη θέση {Search(numbers, target)}.");
        Console.WriteLine($"Αναζήτηση 3: Το στοιχείο {target} βρέθηκε στη θέση {Search(numbers, target)}.");
    }
}
```

```
}  
}
```

Λύση: Η μέθοδος Search χρησιμοποιεί τη μεταβλητή lastFoundIndex για να θυμάται την τελευταία θέση που βρήκε το στοιχείο. Στην πρώτη αναζήτηση, ξεκινά από την αρχή του πίνακα. Στις επόμενες αναζητήσεις, ξεκινά από τη θέση μετά την τελευταία βρέθηκε το στοιχείο, αν αυτό βρέθηκε προηγούμενος.

Άσκηση 2: Εύρεση Πρώτης Εμφάνισης και Επαναλαμβανόμενη Αναζήτηση

Περιγραφή: Δημιούργησε μια κλάση που να περιλαμβάνει μια μέθοδο που θα ψάχνει για ένα στοιχείο σε έναν πίνακα και θα επιστρέφει τη θέση του. Αν το στοιχείο βρεθεί, η επόμενη αναζήτηση θα ξεκινά από τη θέση μετά την τελευταία θέση που βρέθηκε.

Κώδικας:

```
using System;
```

```
public class OptimizedSearch  
{  
    private static int lastFoundIndex = -1;  
  
    public static int FindElement(int[] array, int target)  
    {  
        int startIndex = lastFoundIndex + 1;  
  
        for (int i = startIndex; i < array.Length; i++)  
        {  
            if (array[i] == target)  
            {  
                lastFoundIndex = i;  
                return i;  
            }  
        }  
  
        return -1; // Το στοιχείο δεν βρέθηκε  
    }  
}
```

```

    }

    public static void Main()
    {
        int[] numbers = { 10, 20, 30, 40, 30, 50, 60, 30 };
        int target = 30;

        Console.WriteLine($"Βρίσκω πρώτη εμφάνιση: Το  
στοιχείο {target} βρέθηκε στη θέση {FindElement(numbers,  
target}).");

        Console.WriteLine($"Βρίσκω ξανά: Το στοιχείο {target}  
βρέθηκε στη θέση {FindElement(numbers, target}).");

        Console.WriteLine($"Βρίσκω ξανά: Το στοιχείο {target}  
βρέθηκε στη θέση {FindElement(numbers, target}).");
    }
}

```

Λύση: Ο κώδικας αποθηκεύει την τελευταία θέση όπου βρέθηκε το στοιχείο target. Στις επόμενες αναζητήσεις, η αναζήτηση ξεκινά από τη θέση μετά την τελευταία θέση που βρέθηκε το στοιχείο, κάνοντάς την πιο γρήγορη αν το στοιχείο εμφανίζεται κοντά στην προηγούμενη θέση.

Άσκηση 3: Αναζήτηση και Ενημέρωση Θέσης

Περιγραφή: Δημιούργησε μια κλάση που θα περιλαμβάνει μια μέθοδο για αναζήτηση στοιχείων με βελτιστοποίηση. Αν βρείτε το στοιχείο, πρέπει να επιστρέψετε την τρέχουσα θέση του και να ενημερώσετε την τελευταία θέση που βρέθηκε.

Κώδικας:

```

using System;

public class OptimizedSearchExample
{
    private static int lastIndex = -1;

    public static int FindAndUpdate(int[] array, int target)
    {
        int start = lastIndex + 1;

        for (int i = start; i < array.Length; i++)

```

```

    {
        if (array[i] == target)
        {
            lastIndex = i;
            return i;
        }
    }

    return -1; // Το στοιχείο δεν βρέθηκε
}

public static void Main()
{
    int[] data = { 7, 14, 21, 28, 35, 42, 49 };
    int searchValue = 21;

    Console.WriteLine($"Αναζήτηση 1: Το στοιχείο {searchValue} βρέθηκε στη θέση {FindAndUpdate(data, searchValue)}.");
    Console.WriteLine($"Αναζήτηση 2: Το στοιχείο {searchValue} βρέθηκε στη θέση {FindAndUpdate(data, searchValue)}.");
    Console.WriteLine($"Αναζήτηση 3: Το στοιχείο {searchValue} βρέθηκε στη θέση {FindAndUpdate(data, searchValue)}.");
}
}

```

Λύση: Ο κώδικας αναζητά το στοιχείο από την τελευταία θέση που βρέθηκε, αν η τελευταία θέση είναι γνωστή. Ενημερώνει την τελευταία θέση αν βρει το στοιχείο, κάνοντάς την αναζήτηση πιο αποδοτική στις επόμενες κλήσεις.

Αυτές οι ασκήσεις είναι καλές για να εξασκηθείς στη Βελτιστοποιημένη Γραμμική Αναζήτηση και να κατανοήσεις πώς μπορείς να κάνεις την αναζήτηση πιο γρήγορη με αποθήκευση της τελευταίας θέσης που βρέθηκε το στοιχείο.