

Άσκηση 1: Βασική Αναζήτηση με Άλματα

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί την αναζήτηση με άλματα για να βρει έναν αριθμό σε έναν ταξινομημένο πίνακα αριθμών. Εκτύπωσε τη θέση του αριθμού ή ένα μήνυμα αν δεν βρεθεί.

Λύση:

```
using System;

class JumpSearchExample
{
    static void Main()
    {
        int[] array = { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 };
        int target = 15;

        int result = JumpSearch(array, target);
        if (result != -1)
        {
            Console.WriteLine($"Ο αριθμός {target} βρέθηκε στην  
θέση {result}.");
        }
        else
        {
            Console.WriteLine($"Ο αριθμός {target} δεν  
βρέθηκε.");
        }
    }

    static int JumpSearch(int[] array, int target)
    {
        int length = array.Length;
        int jump = (int)Math.Sqrt(length); // Καθορίστε το  
μέγεθος του άλματος
        int prev = 0;

        // Βρείτε το μπλοκ όπου μπορεί να  
βρίσκεται το target
        while (array[Math.Min(jump, length) - 1] < target)
        {
            prev = jump;
        }
    }
}
```

```

        jump += (int)Math.Sqrt(length);
        if (prev >= length)
            return -1;
    }

    // Κοιτάξτε μέσα στο μπλοκ που βρήκατε
    for (int i = prev; i < Math.Min(jump, length); i++)
    {
        if (array[i] == target)
            return i;
    }

    return -1; // Εάν δεν βρείτε το target
}

```

Άσκηση 2: Εύρεση Ελάχιστης Θέσης

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί την αναζήτηση με άλματα για να βρει την ελάχιστη θέση ενός αριθμού που εμφανίζεται σε έναν ταξινομημένο πίνακα. Αν ο αριθμός εμφανίζεται περισσότερες από μία φορές, επιστρέφει τη θέση της πρώτης εμφάνισης.

Λύση:

```

using System;

class JumpSearchExample
{
    static void Main()
    {
        int[] array = { 1, 2, 2, 2, 3, 4, 5, 6, 7, 8, 9 };
        int target = 2;

        int result = JumpSearch(array, target);
        if (result != -1)
        {
            Console.WriteLine($"Ο αριθμός {target} βρέθηκε στη
θέση {result}.");
        }
    }
}

```

```

        else
        {
            Console.WriteLine($"Ο αριθμός {target} δεν
βρέθηκε.");
        }
    }

    static int JumpSearch(int[] array, int target)
    {
        int length = array.Length;
        int jump = (int)Math.Sqrt(length);
        int prev = 0;

        while (array[Math.Min(jump, length) - 1] < target)
        {
            prev = jump;
            jump += (int)Math.Sqrt(length);
            if (prev >= length)
                return -1;
        }

        for (int i = prev; i < Math.Min(jump, length); i++)
        {
            if (array[i] == target)
                return i;
        }

        return -1;
    }
}

```

Άσκηση 3: Δοκιμή με Ψευδές Αριθμό

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί την αναζήτηση με άλματα για να δοκιμάσεις αν μπορείς να βρεις έναν αριθμό που δεν υπάρχει στον πίνακα. Εκτύπωσε το μήνυμα που δείχνει ότι δεν βρέθηκε.

Λύση:

```
using System;
```

```
class JumpSearchExample
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int[] array = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 };
```

```
        int target = 55; // Δ ε ν υ π ά ρ χ ε ι σ τ ο ν π ί ν α κ α
```

```
        int result = JumpSearch(array, target);
```

```
        if (result != -1)
```

```
        {
```

```
            Console.WriteLine($"Ο α ρ ι θ μ ό ς {target} β ρ έ θ η κ ε σ τ η  
θ έ σ η {result}.");
```

```
        }
```

```
        else
```

```
        {
```

```
            Console.WriteLine($"Ο α ρ ι θ μ ό ς {target} δ ε ν  
β ρ έ θ η κ ε.");
```

```
        }
```

```
    }
```

```
    static int JumpSearch(int[] array, int target)
```

```
    {
```

```
        int length = array.Length;
```

```
        int jump = (int)Math.Sqrt(length);
```

```
        int prev = 0;
```

```
        while (array[Math.Min(jump, length) - 1] < target)
```

```
        {
```

```
            prev = jump;
```

```
            jump += (int)Math.Sqrt(length);
```

```
            if (prev >= length)
```

```
                return -1;
```

```
        }
```

```
        for (int i = prev; i < Math.Min(jump, length); i++)
```

```
        {
```

```
            if (array[i] == target)
```

```
                return i;
```

```

    }

    return -1;
}
}

```

Άσκηση 4: Δοκιμή με Μεγαλύτερο Πίνακα

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί την αναζήτηση με άλματα για να βρεις έναν αριθμό σε έναν μεγαλύτερο ταξινομημένο πίνακα. Δοκίμασε διάφορους αριθμούς και εκτύπωσε τα αποτελέσματα.

Λύση:

```

using System;

class JumpSearchExample
{
    static void Main()
    {
        int[] array = { 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70,
75, 80, 85, 90, 95, 100 };
        int[] targets = { 25, 75, 100, 105 };

        foreach (int target in targets)
        {
            int result = JumpSearch(array, target);
            if (result != -1)
            {
                Console.WriteLine($"Ο αριθμός {target} βρέθηκε
στη θέση {result}.");
            }
            else
            {
                Console.WriteLine($"Ο αριθμός {target} δεν
βρέθηκε.");
            }
        }
    }
}

```

```
    }  
}
```

```
static int JumpSearch(int[] array, int target)  
{  
    int length = array.Length;  
    int jump = (int)Math.Sqrt(length);  
    int prev = 0;  
  
    while (array[Math.Min(jump, length) - 1] < target)  
    {  
        prev = jump;  
        jump += (int)Math.Sqrt(length);  
        if (prev >= length)  
            return -1;  
    }  
  
    for (int i = prev; i < Math.Min(jump, length); i++)  
    {  
        if (array[i] == target)  
            return i;  
    }  
  
    return -1;  
}
```