

Бинарна претрага низа

Бинарна претрага је алгоритам за претрагу елемента унутар сортираног низа.

Постављају се две промењиве, startIndeks и krajIndeks на почетну и крајњу позицију унутар низа.

Пронађе се позиција средњег броја у низу и ако је то тражени број, претрага је завршена.

Ако то није тражени број, проверава се вредност средњег броја, да ли је већа или мања од вредности траженог броја.

Ако је већа, тражени број је у доњој половини низа и startIndeks и krajIndeks се постављају на почетну и позицију на средини низа.

Ако је мања, тражени број је у горњој половини низа и startIndeks и krajIndeks се постављају на позицију на средини низа и на крајњу позицију.

Овакава подела се врши све док се не пронађе тражени број или док се не утврди да број не постоји у низу.

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] niz = { 1, 4, 7, 9, 16, 56, 70 };
            int n = niz.Length, element = 16, sredina;
            int startIndeks = 0, krajIndeks = n - 1,
                pronadjenIndeks = 0;
            while (startIndeks <= krajIndeks)
            {
                sredina = startIndeks + (krajIndeks - startIndeks)
                    / 2;
                if (niz[sredina] == element)
                {
                    pronadjenIndeks = sredina;
                    break;
                }
                else if (niz[sredina] < element)
                {
                    startIndeks = sredina + 1;
                    pronadjenIndeks = -1;
                }
                else
                {
                    krajIndeks = sredina - 1;
                    pronadjenIndeks = -1;
                }
            }
            if (pronadjenIndeks == -1)
                Console.WriteLine("Element nije u nizu.");
        }
    }
}
```

```

        else
            Console.WriteLine($"Element pronadjen na poziciji
                               {pronadjenIndeks}.");
    }
}
}

```

### Метода сортирања bubble sort

У низ целих бројева је унето  $n$  елемената у несортираном редоследу.

Да би се сортирали бројеви у растућем редоследу, упоређују се кроз две петље.

Спољна петља итерира од индекса 0 до индекса  $n$  а унутрашња од  $i + 1$  до  $n$ , са тиме што никада неће упоређивати бројеве на истим позицијама.

До замене позиција долази ако је већи број у спољној петљи.

```

using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] niz = new int[50];
            int i, j, duzinaNiza, x;
            Console.Write("Uneti duzinu niza: ");
            duzinaNiza = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Uneti elemente niza: ");
            for (i = 0; i < duzinaNiza; i++)
            {
                niz[i] = Convert.ToInt32(Console.ReadLine());
            }
            for (i = 0; i < duzinaNiza; i++)
            {
                for (j = 0; j < (duzinaNiza - i - 1); j++)
                {
                    if (niz[j] > niz[j + 1])
                    {
                        x = niz[j];
                        niz[j] = niz[j + 1];
                        niz[j + 1] = x;
                    }
                }
            }
            Console.WriteLine("Niz sortiran bubble sort metodom: ");
            for (i = 0; i < duzinaNiza; i++)
            {
                Console.Write(niz[i] + " ");
            }
        }
    }
}

```

```

        Console.WriteLine("\n");
    }
}

```

#### Метода ротирања елемената низа

Ротирање елемената низа значи да елементи низа размењују вредности без губљења вредности, додавања нових вредности или њихових модификација.

Елементи низа се могу ротирати вредности у једном од два смера: у леву страну или у десну страну.

Ротирање у десно значи да вредност последњег елемента у низу постаје вредност 1.елемента у низу, вредност 1.елемента у низу постаје вредност 2.елемента у низу...

Ротирање у лево значи да вредност првог елемента у низу постаје вредност последњег елемента у низу, вредност 2.елемента у низу постаје вредност 1.елемента у низу...

Варијације алгоритма ротирања зависе од корака прескакања броја елемената током ротирања.

Пример: ротирати вредности у низу за једно место на десну страну

```

using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] niz = { 1, 4, 7, 9, 16, 56, 70 };
            int i, pom, duzinaNiza = niz.Length;
            pom = niz[duzinaNiza - 1];
            for (i = duzinaNiza - 1; i > 0; i--)
            {
                niz[i] = niz[i - 1];
            }
            niz[0] = pom;
            for (i = 0; i < duzinaNiza; i++)
            {
                Console.Write(niz[i] + " ");
            }
            Console.WriteLine("\n");
        }
    }
}

```