

Programiranje – III razred

Sortiranje nizova

Sortiranje nizova



Sortirati niz znači poređati njegove članove u nekom (rastućem, opadajućem, strogo rastućem, strogo opadajućem) poretku. Brojevi se sortiraju u odnosu na relaciju $< i >$, a znaci i reči u odnosu na engleski alfabet, zahvaljujući činjenici da su već predstavljeni pomoću ASCII koda u obliku brojeva.

Primer nesortiranog niza:

$A[5]=\{4, 3, 10, -5, 121\}$

Isti niz nakon sortiranja po rastućem redosledu:

$A[5]=\{-5, 3, 4, 10, 121\}$

Isti niz nakon sortiranja po opadajućem redosledu:

$A[5]=\{121, 10, 4, 3, -5\}$

Sortiranje nizova



Kako ljudi sortiraju niz?

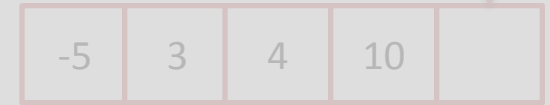
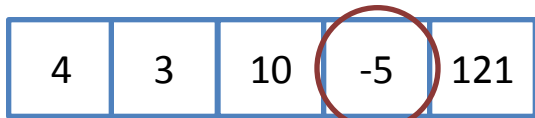
Najčešći način na koji ljudi sortiraju niz je sledeći: Ukoliko niz treba sortirati po rastućem redosledu (od najmanjeg ka najvećem članu), pronade se najmanji član, precrtava u nizu i upiše kao prvi element novog, sortiranog niza.

Postupak se ponavlja, tražeći novi najmanji član niza (prethodni najmanji član je precrtan).

Ovaj se proces ponavlja dok u nizu ne ostane samo jedan element koji se prepíše kao poslednji član novog niza.

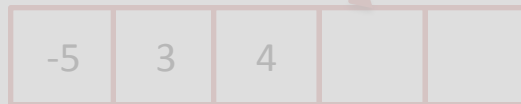
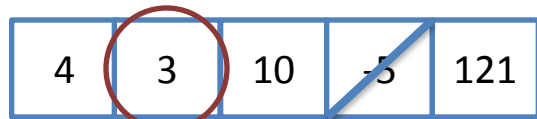
Sortiranje nizova

Kako ljudi sortiraju niz?



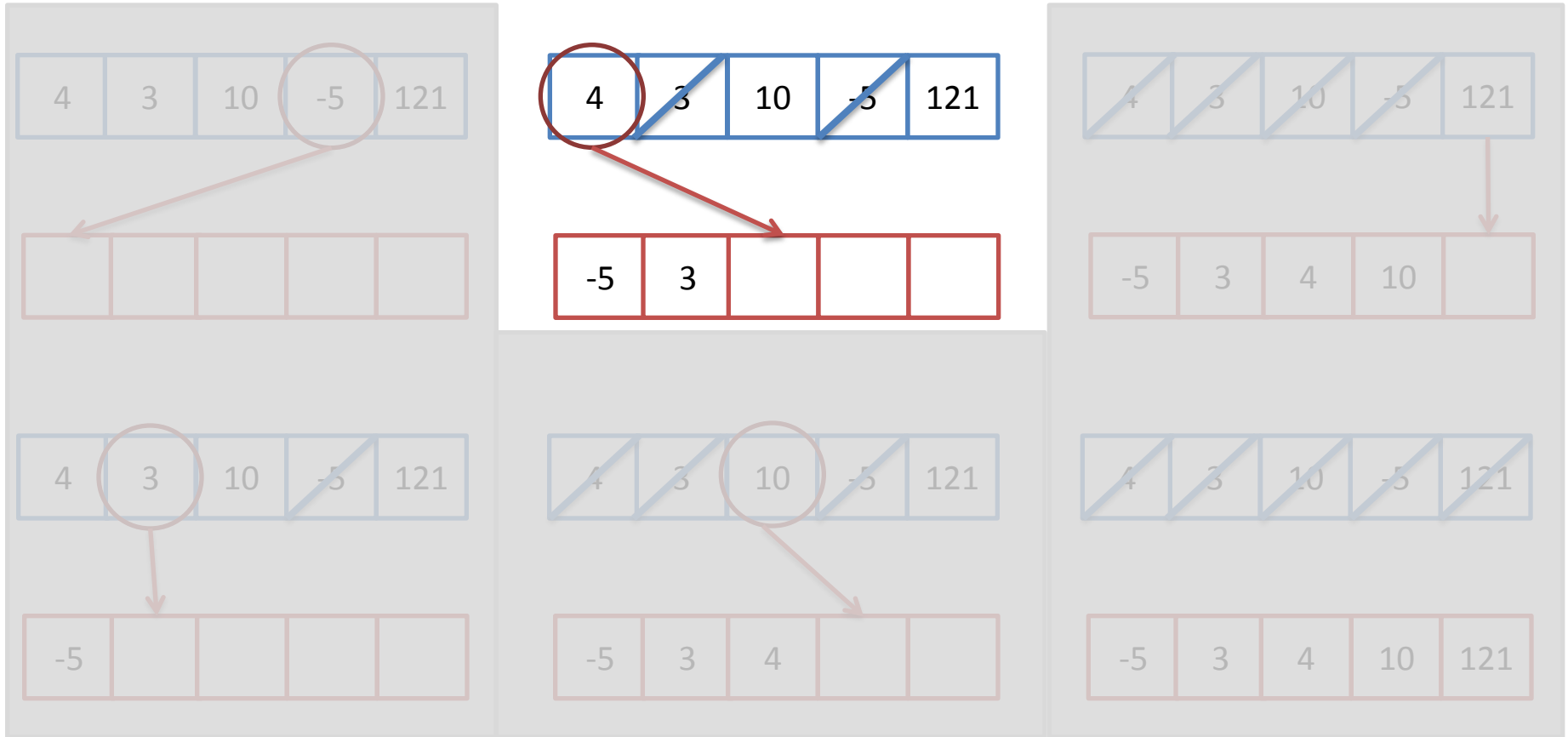
Sortiranje nizova

Kako ljudi sortiraju niz?



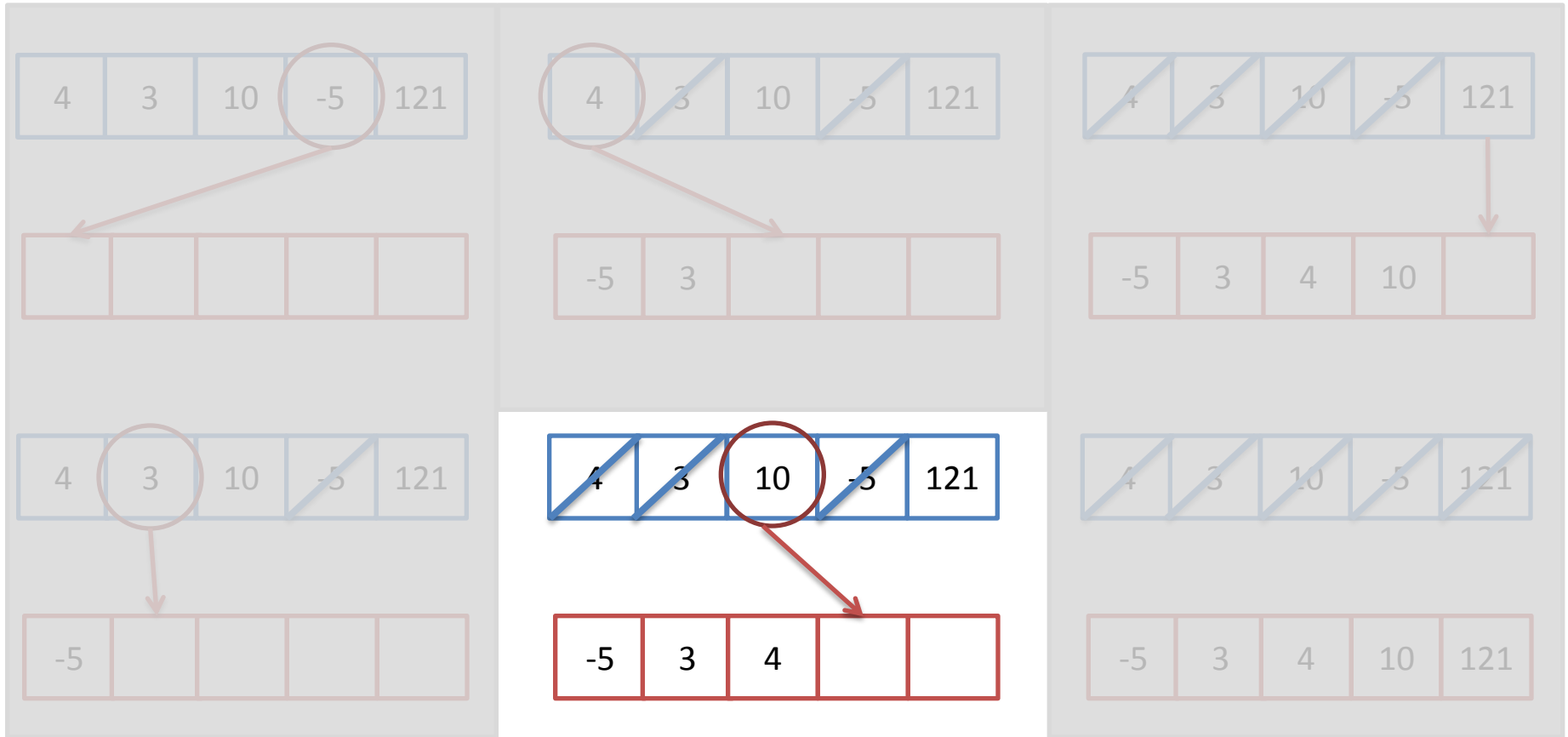
Sortiranje nizova

Kako ljudi sortiraju niz?



Sortiranje nizova

Kako ljudi sortiraju niz?



Sortiranje nizova

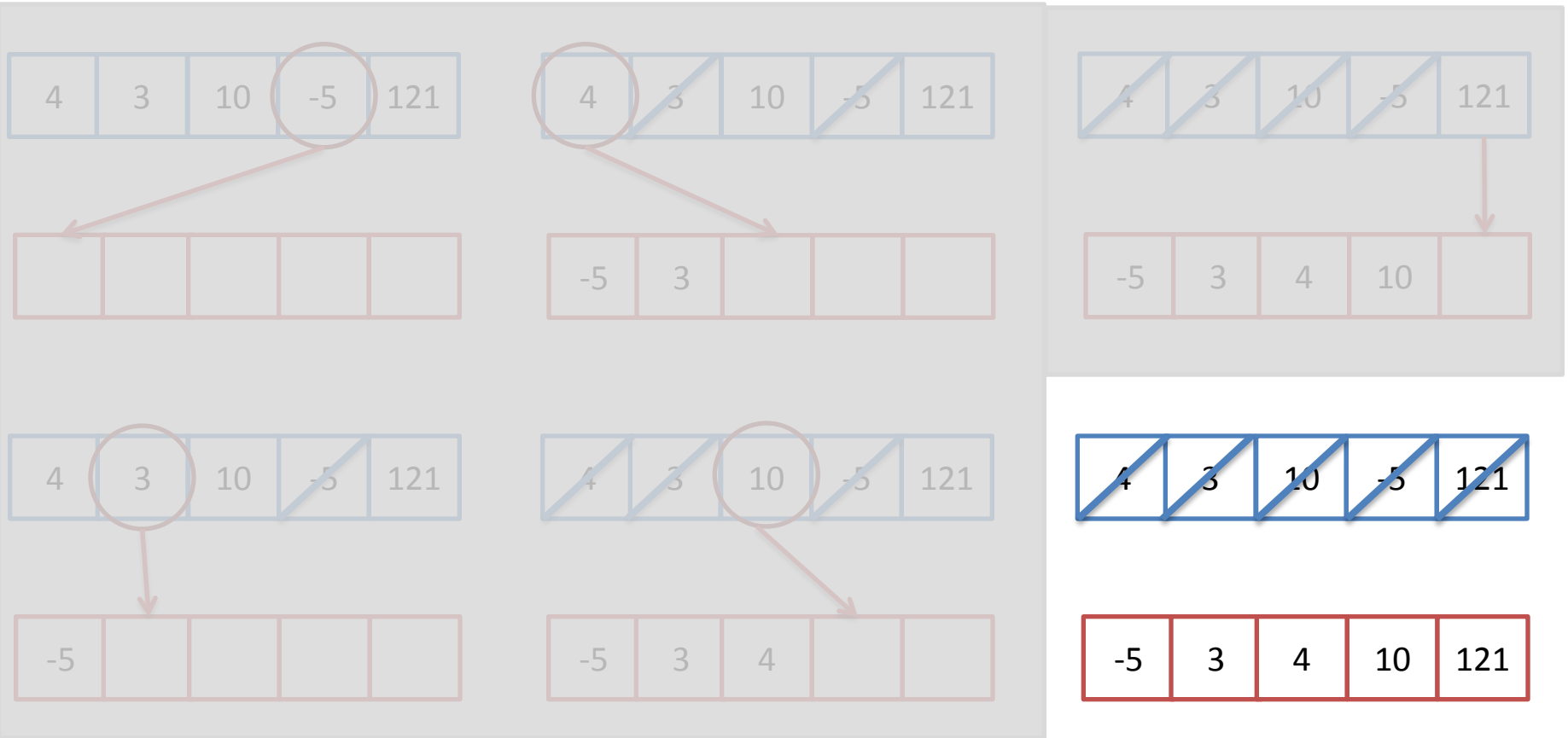


Kako ljudi sortiraju niz?



Sortiranje nizova

Kako ljudi sortiraju niz?



Sortiranje nizova – algoritmi sortiranja



Podela algoritama se vrši na osnovu globalnog metoda sortiranja i čine je dve grupe:

- Algoritmi sortiranja poređenjem
- Algoritmi sortiranja bez poređenja

Algoritmi sortiranja poređenjem:

Kod ove grupe algoritama sortiranja, elementi ulaznog skupa se preraspoređuju na osnovu nekog ključa. Ključ predstavlja posebno odabranu vrednost koja određuje da li je neki element veći, manji ili jednak nekom drugom elementu ulaznog skupa. Najveći broj algoritama pripada ovoj grupi, a oni se dalje mogu klasifikovati na sledeći način:

1. Algoritmi umetanja

- Insertion sort
- Shell sort

2. Algoritmi izbora

- Selection sort
- Heapsort

3. Algoritmi zamene

- Bubble sort
- Quicksort

Sortiranje nizova – algoritmi sortiranja



Algoritmi sortiranja bez poređenja:

Ovi algoritmi koriste druge metode osim poređenja ključeva elemenata da bi došli do željenog redosleda elemenata.

Osnovni razlog za razvoj ovih algoritama je što su algoritmi sortiranja poređenjem generalno znatno sporiji.

Najznačajniji algoritmi iz ove grupe su:

- Bucket sort
- Counting sort
- Radix sort

Slučajno generisan broj



Slučajno generisani brojevi:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
namespace RandomNumber
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            //Način kreiranja slučajnog broja u željenom opsegu
```

```
            Random rnd = new Random(); //Kreiranje instance klase Random
```

```
            int zeljeniBroj = rnd.Next(1, 100); // Generisan je slučajan broj od 0 do 99
```

```
            int kocka = rnd.Next(1,7); //Simulacija kocke - generisan slučajan broj od 1 do 6
```

```
                Console.WriteLine("Prvi broj je: " + zeljeniBroj);
```

```
                Console.WriteLine("Kocka je pala na..." + kocka);
```

```
                Console.ReadLine();
```

```
        }
```

```
    }
```

```
}
```

Napomena:

Slučajno generisani brojevi su potrebni radi lakšeg punjenja nizova elementima.

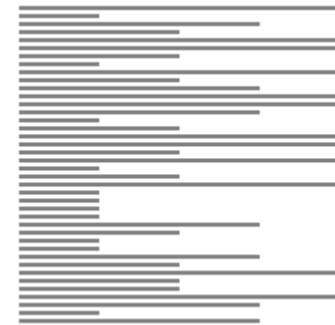
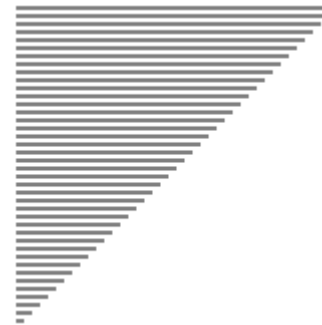
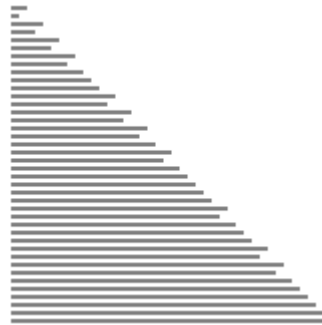
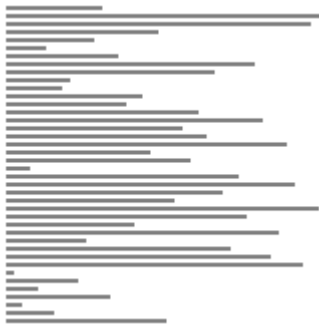
Sortiranje nizova – Insertion sort



Sortiranje umetanjem (Insertion sort)

Pri sortiranju umetanjem, sortirani deo se povećava tako što se prvi element nesortiranog dela „umetne“ na odgovarajuće mesto u sortirani deo.

6 5 3 1 8 7 2 4



Sortiranje nizova – Insertion sort



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace InsertSort
{
    class Program
    {
        static void Main(string[] args)
        {
            //Deklaracija niza od 10 elemenata
            int[] x = new int[10];
            Random rnd = new Random();
            //Punjenje niza slucajnim brojevima
            for (int i = 0; i < x.Length; i++)
            {
                x[i] = rnd.Next(1,100);
            }
            //Prikaz na konzoli
            Console.WriteLine("Elementi nesortiranog niza su: ");
            for (int i = 0; i < x.Length; i++)
            {
                Console.WriteLine("array[" + i + "] = " + x[i]);
            }
            Console.ReadLine();
        }
    }
}
```

```
//S O R T I R A N J E

int temp, k;
for (int i = 1; i < x.Length; i++)
{
    temp = x[i];
    k = i - 1;
    while (k >= 0 && x[k] > temp)
    {
        x[k + 1] = x[k];
        k--;
    }
    x[k + 1] = temp;
}
//Prikaz na konzoli
Console.WriteLine("Elementi SORTIRANOG niza su: ");
for (int i = 0; i < x.Length; i++)
{
    Console.WriteLine("array[" + i + "] = " + x[i]);
}
Console.ReadLine();
}
```

Sortiranje nizova – Insertion sort



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
namespace InsertSort
```

```
{
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

```
//Deklaracija
```

```
int[] x =
```

```
Random rnd =
```

```
//Punjenje
```

```
for (int i = 0; i < x.Length; i++)
```

```
{
```

```
x[i] =
```

```
}
```

```
//Prikaz
```

```
Console.WriteLine("Elementi nesortiranog niza su: ");
```

```
for (int i = 0; i < x.Length; i++)
```

```
{
```

```
Console.WriteLine("array[" + i + "] = " + x[i]);
```

```
}
```

```
Console.ReadLine();
```

```
}
```

```
//SORTIRANJE
```

```
int temp, k;
```

```
for (int i = 1; i < x.Length; i++)
```

```
{
```

```
temp = x[i];
```

```
k = i;
```

```
while (k > 0 && x[k] < temp)
```

```
{
```

```
x[k + 1] = x[k];
```

```
k--;
```

```
x[k + 1] = temp;
```

```
}
```

```
}
```

```
Console.WriteLine("Elementi SORTIRANOG niza su: ");
```

```
for (int i = 0; i < x.Length; i++)
```

```
{
```

```
Console.WriteLine("array[" + i + "] = " + x[i]);
```

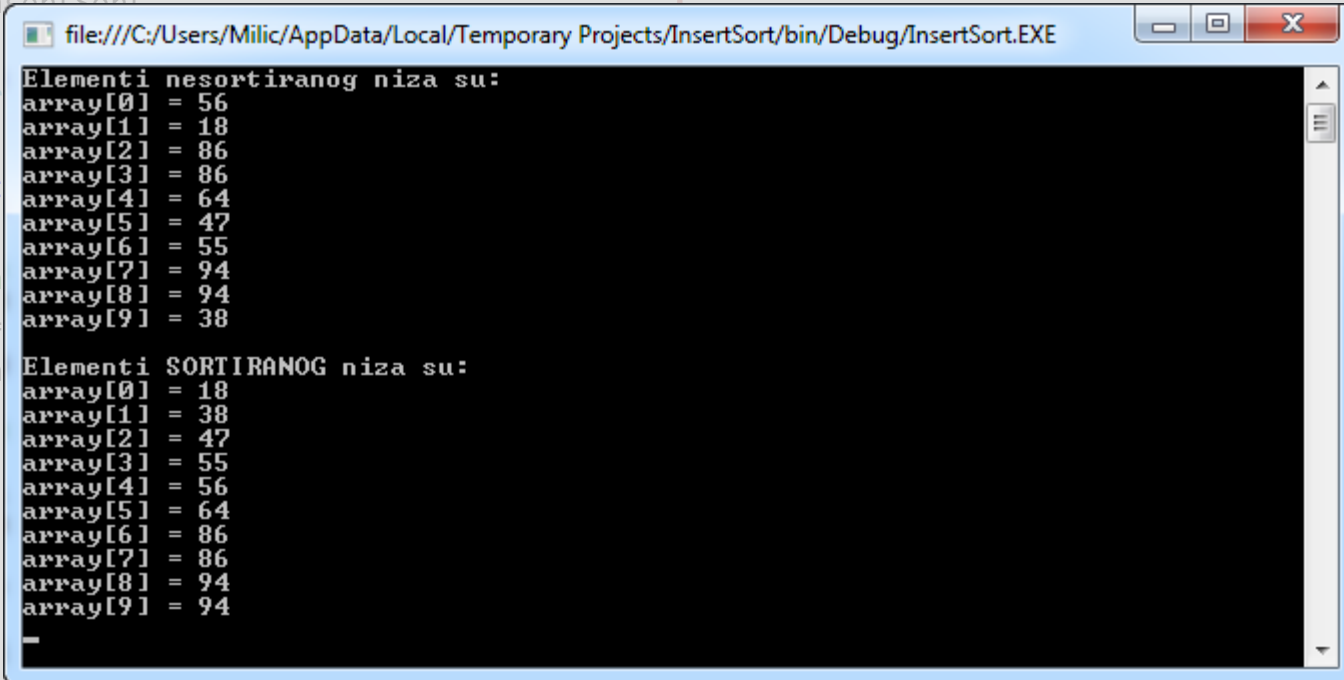
```
}
```

```
Console.ReadLine();
```

```
}
```

```
}
```

```
}
```

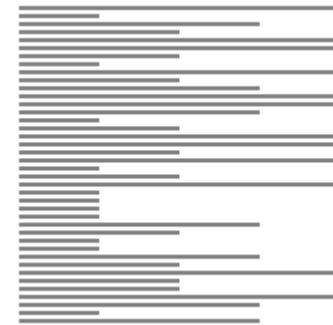
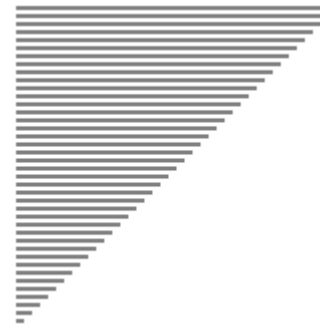
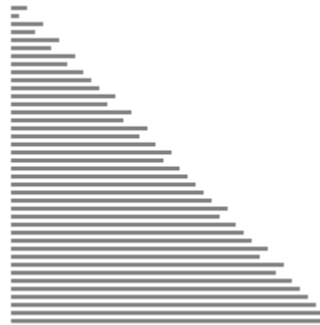
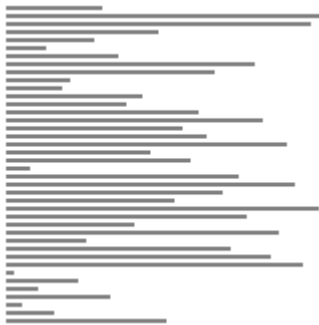


Sortiranje nizova – Bubble sort



Bubble sort je prilično neefikasan algoritam, ali se često primenjuje zbog svoje jednostavnosti. Njegov princip je prolaz kroz ulazni skup više puta, pri čemu se upoređuju dva susedna elementa i ako je prvi veći od drugog, oni menjaju mesta. Algoritam se može malo poboljšati ako primetimo da su nakon i-tog prolaska i najvećih elemenata sortirani i da njih ne treba dalje proveravati.

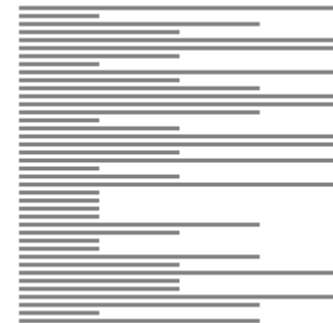
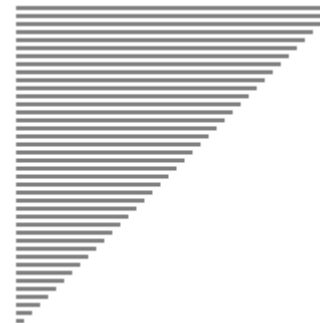
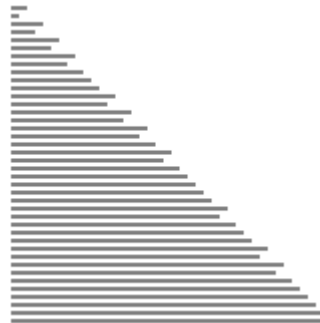
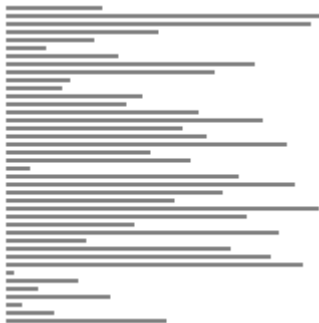
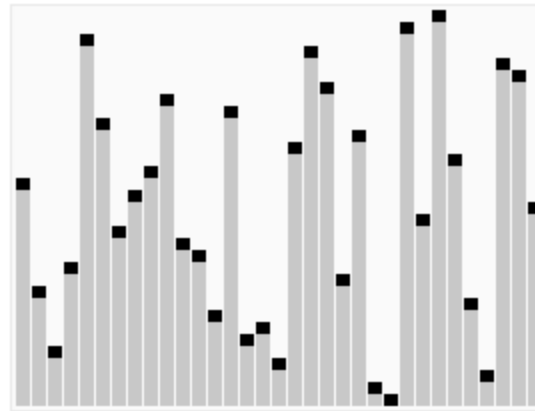
6 5 3 1 8 7 2 4



Sortiranje nizova – Quick sort



Ovo je najčešće upotrebljavan algoritam sortiranja. Nije težak za implementaciju, a koristi manje resursa (vremena i prostora) nego bilo koji drugi algoritam sortiranja, u većini slučajeva. Algoritam ne zahteva dodatnu memoriju, samo $n \cdot \log(n)$ operacija u proseku za sortiranje n elemenata, i ima izuzetno kratku unutrašnju petlju. Loše strane algoritma su što je rekurzivan (nerekurzivna varijanta je mnogo složenija). Postoje i verzije ovog algoritma koje ga poboljšavaju.



Zadaci



1. Kreirati program u kome se nakon unosa dužine niza, elementi niza popunjavaju slučajnim brojevima i nakon toga prikazuju na konzoli.
2. Proširiti prethodni zadatak tako da se nakon ispisa elemenata niza, niz sortira, pa se taj, sortirani niz, prikaže preko konzole.

Programiranje – III razred

Sortiranje nizova
