

Претраживање низова

Низови података могу бити урађени по растућем или опадајућем редоследу вредности елемената.

Низ је уређен по растућем редоследу ако је сваки наредни елемент већи од претходног а у опадајућем редоследу ако је сваки елемент мањи од претходног.

Постоје и многе технике претраживања и уређивања низова.

Поступак не зависи од типа елемената низа.

Најједноставнији начин да се утврди да ли нека вредност постоји у низу јесте да се испитује сваки елемент.

Претраживање се завршава онда када се пронађе тражени елемент.

Поступак се зове **секвенцијално претраживање**.

Да је тражена вредност пронађена зна се по томе што је циклус претраживања завршен пре краја низа.

Ако је елемент пронађен, вредност функције је различита од нуле.

Пример 01: У низу целих бројева одредити да ли постоји број 1.

```
//u nizu celih brojeva odrediti da li postoji broj 1
```

```
//i koliko se puta pojavljuje u nizu
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a[100], n, i, b = 0;
```

```
    cout << "Unesi duzinu niza: ";
```

```
    cin >> n;
```

```
    cout << "Unesi elemente niza:" << endl;
```

```
    for (i = 0; i < n; i++) cin >> a[i];
```

```
    for (i = 0; i < n; i++) if (a[i] == 1) b++;
```

```
    if (b > 0) cout << "U nizu postoji broj 1 i pojavljuje se "
```

```
        << b << " puta." << endl;
```

```
    else cout << "U nizu ne postoji broj 1." << endl;
```

```
    return 0;
```

```
}
```

```
Unesi duzinu niza: 8
Unesi elemente niza:
0 0 0 1 1 0 6 -2
U nizu postoji broj 1 i pojavljuje se 2 puta.
```

Пример 02: Одредити да ли постоји тражени број у низу целих бројева.

```
//u nizu celih brojeva odrediti da li postoji trazeni broj
```

```
//i koliko se puta pojavljuje u nizu
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a[100], n, i, x;
```

```
    bool nadjen = false;
```

```
    cout << "Unesi duzinu niza: ";
```

```
    cin >> n;
```

```
    cout << "Unesi elemente niza:" << endl;
```

```
    for (i = 0; i < n; i++) cin >> a[i];
```

```
    cout << "Uneti trazeni ceo broj: ";
```

```
    cin >> x;
```

```
    for (i = 0; i < n; i++) if (a[i] == x) nadjen = true;
```

```
    if (nadjen) cout << "U nizu postoji broj " << x << endl;
```

```
    else cout << "U nizu ne postoji broj " << x << endl;
```

```
    return 0;
```

```
}
```

```
Unesi duzinu niza: 3
Unesi elemente niza:
1 2 3
Uneti trazeni ceo broj: 2
U nizu postoji broj 2
```

Пример 03: Одредити да ли постоје бројеви у траженом опсегу целих бројева.

```
//u nizu celih brojeva odrediti da li postoje trazeni brojevi
//koji se pojavljuju u nekom opsegu i koliko se puta pojavljuju u nizu
#include<iostream>
using namespace std;
```

```
int main()
{
    int a[100], n, i, x, y, b = 0;
    cout << "Unesi duzinu niza: ";
    cin >> n;
    cout << "Unesi elemente niza:" << endl;
    for (i = 0; i < n; i++) cin >> a[i];
    cout << "Uneti donju i gornju granicu opsega: ";
    cin >> x >> y;
    for (i = 0; i < n; i++)
        if ((a[i] >= x) && (a[i] <= y)) b++;
    if (b != 0)
    {
        cout << "U nizu postoje brojevi u opsegu od "
              << x << " do " << y << "." << endl;
        cout << "Brojevi iz ovog opsega se pojavljuju "
              << b << " puta." << endl;
    }
    else cout << "U nizu ne postoji brojevi iz zadatog opsega." << endl;
    return 0;
}
```

```
Unesi duzinu niza: 5
Unesi elemente niza:
3 4 6 0 2
Uneti donju i gornju granicu opsega: -1 2
U nizu postoje brojevi u opsegu od -1 do 2.
Brojevi iz ovog opsega se pojavljuju 2 puta.
```

### Сортирање низова

Сортирање (уређење) низа захтева алгоритам за упоређивање једног елемента низа са осталим елементима.

Ако се испуни тражени услов за сортирање, потребно је извршити замену позиција елемента низа.

Алгоритми (методе) за сортирање захтевају више пролаза при сортирању.

У сваком пролазу се уочи неки елемент који се упоређује са осталим елементима.

Алгоритми за сортирање низова су:

1. метода избора
2. метода уметања
3. метода замене суседа
4. метода поделе

### Метода избора

Метода се заснива на избору најмањег елемента низа и довођење на прво место у низу, затим се налази следећи најмањи елемент у низу и доводи на следеће место у низу итд.

Метода захтева две петље: прва креће од 0 и иде до n-1 елемента, друга почиње од другог елемента и иде до краја низа.

Други елемент има другачију вредност индекса за сваки циклус.

Пример 01: Сортирање елемента од највећег до најмањег елемента.

```
//sortiranje od najveceg do najmanjeg elementa
#include<iostream>
using namespace std;

int main()
{
    int a[100], n, i, j, temp;
    cout << "Uneti broj elemenata niza: ";
    cin >> n;
    cout << "Uneti elemente niza: " << endl;
    for (i = 0; i < n; i++) cin >> a[i];
    for(i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
        {
            if (a[i] < a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    for (i = 0; i < n; i++) cout << a[i] << " ";
    cout << endl;
    return 0;
}
```

---

```
Uneti broj elemenata niza: 5
Uneti elemente niza:
4 5 1 3 2
5 4 3 2 1
```

Пример 02: Сортирање низа од најмањег до највећег елемента

```
//sortiranje od najmanjeg do najveceg elementa
#include<iostream>
using namespace std;

int main()
{
    int a[100], n, i, j, temp;
    cout << "Uneti broj elemenata niza: ";
    cin >> n;
    cout << "Uneti elemente niza: " << endl;
    for (i = 0; i < n; i++) cin >> a[i];
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
        {
            if (a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    for (i = 0; i < n; i++) cout << a[i] << " ";
    cout << endl;
    return 0;
}
```

---

```
Uneti broj elemenata niza: 5
Uneti elemente niza:
4 37 12 58 2
2 4 12 37 58
```

Сортирање низова

Пример 01: Написати програм који претражује или сортира низ.

```
#include <iostream>
using namespace std;
void potraga(int a[], int);
void sort_raste(int a[], int);
void sort_opada(int a[], int);
void unos(int a[], int);
void prikaz(int a[], int);

int main()
{
    int a[100], n, x;
    bool izlaz = false;
    cout << "Koliko elemenata ima niz? ";
    cin >> n;
    unos(a, n);
    cout << "Nesortiran niz: ";
    prikaz(a, n);
    cout << endl;
    while (!izlaz)
    {
        cout << "Izabrati opciju:" << endl;
        cout << "1 - pretraga niza" << endl;
        cout << "2 - sortiranje niza po rastucem redosledu" << endl;
        cout << "3 - sortiranje niza po opadajucem redosledu" << endl;
        cout << "4 - izlaz iz programa" << endl;
        cin >> x;
        switch (x)
        {
            case 1: potraga(a, n); break;
            case 2: sort_raste(a, n); break;
            case 3: sort_opada(a, n); break;
            case 4: izlaz = true; break;
            default: cout << "Pogresan unos" << endl;
        }
        cout << endl;
    }
    return 0;
}

void potraga(int a[], int n)
{
    int c = 0, m;
    cout << "Uneti vrednost za pretragu: ";
    cin >> m;
    for (int i = 0; i < n; i++)
        if (a[i] == m) c++;
    if (c != 0) cout << "Pronadjeno je elemenata: " << c << endl;
    else cout << "Nijedan od trazених elemenata nije pronadjен." << endl;
}

void sort_raste(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (a[i] > a[j])
            {
```

```
                int x = a[i];
                a[i] = a[j];
                a[j] = x;
            }
        cout << "Sortiran rastuci niz: ";
        prikaz(a, n);
    }

void sort_opada(int a[], int n)
{
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (a[i] < a[j])
            {
                int x = a[i];
                a[i] = a[j];
                a[j] = x;
            }
        cout << "Sortiran opadajuci niz: ";
        prikaz(a, n);
}

void unos(int a[], int n)
{
    cout << "Uneti " << n << " elemenata niza\n";
    for (int i = 0; i < n; i++)
    {
        cout << "a[" << i << "]=";
        cin >> a[i];
    }
}

void prikaz(int a[], int n)
{
    for (int i = 0; i < n; i++) cout << a[i] << " ";
    cout << endl;
}
```

```
Koliko elemenata ima niz? 3
Uneti 3 elemenata niza
a[0]=5
a[1]=-1
a[2]=0
Nesortiran niz: 5 -1 0

Izabrati opciju:
1 - pretraga niza
2 - sortiranje niza po rastucem redosledu
3 - sortiranje niza po opadajucem redosledu
4 - izlaz iz programa
1
Uneti vrednost za pretragu: -1
Pronadjeno je elemenata: 1

Izabrati opciju:
1 - pretraga niza
2 - sortiranje niza po rastucem redosledu
3 - sortiranje niza po opadajucem redosledu
4 - izlaz iz programa
2
Sortiran rastuci niz: -1 0 5

Izabrati opciju:
1 - pretraga niza
2 - sortiranje niza po rastucem redosledu
3 - sortiranje niza po opadajucem redosledu
4 - izlaz iz programa
3
Sortiran opadajuci niz: 5 0 -1

Izabrati opciju:
1 - pretraga niza
2 - sortiranje niza po rastucem redosledu
3 - sortiranje niza po opadajucem redosledu
4 - izlaz iz programa
4

Press any key to continue . . . █
```