

Rendező algoritmusok

Tartalomjegyzék

Csere.....	2
Tömbelemek cseréje.....	2
Tömbelemek cseréje ha $a[i] > a[i+1]$	3
$n-1$ csere, ha $a[i] > a[i+1]$	4
$(n-1) * (n-1)$ csere, ha $a[i] > a[i+1]$	5
Buborék rendezés.....	6
Cserés rendezés.....	7
Minimum kiválasztásos rendezés.....	8
Beszúrásos rendezés.....	9

Készítette: Gál Tamás

[Creative Commons](#) -Nevezd meg!-Ne add el!-Így add tovább! 2.5 Magyarország licenc alatt használható

Csere

Mondatszerű leírás:

```
tmp = a  
a = b  
b = tmp
```

C# és Java kód

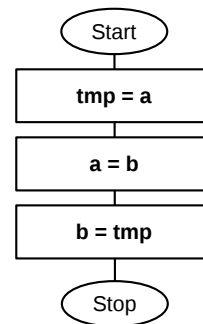
```
int a = 5;  
int b = 6;  
int tmp;
```

```
tmp = a;  
a = b;  
b = tmp;
```

```
Console.WriteLine("a: "+a);  
Console.WriteLine("b: "+b);
```

Java esetén:

- Console.WriteLine helyett System.out.println



Tömbelemek cseréje

Mondatszerű leírás:

```
tmp = a[i]  
a[i] = a[i+1]  
a[i+1] = tmp
```

C# kód

```
int[] a = {22, 5, 4, 2};  
int n = a.Length; //a tömb mérete  
int tmp, i = 0;
```

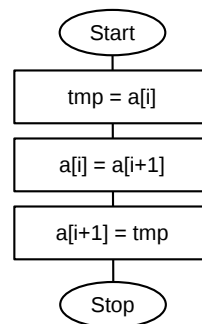
```
for (int j=0;j<n;j++)  
{Console.Write(a[j]+" ")};
```

```
tmp = a[i];  
a[i] = a[i+1];  
a[i+1] = tmp;
```

```
Console.WriteLine();  
for (int j=0;j<n;j++)  
{Console.Write(a[j]+" ")};
```

Java esetén:

- Length helyett length
- Console.Write helyett System.out.print
- Console.WriteLine helyett System.out.println



Tömbelemek cseréje ha $a[i] > a[i+1]$

Csak akkor cserélünk, ha a nagyobb indexű elem értéke kisebb.

Mondatszerű leírás:

```
ha  $a[i] > a[i+1]$  akkor  
  tmp  = a[i]  
  a[i]  = a[i+1]  
  a[i+1] = tmp  
ha vége
```

C# kód

```
int[] a = {22, 5, 2, 4};  
int n = a.Length; //a tömb mérete  
int tmp, i = 0;
```

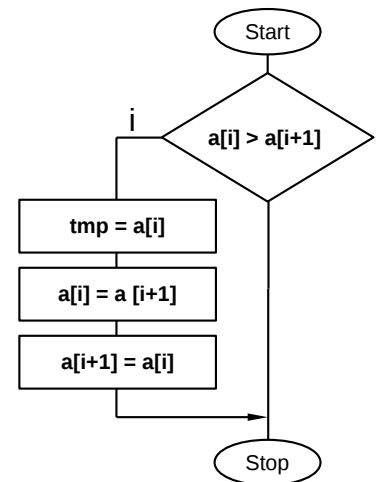
```
for (int j=0; j<n; j++)  
{Console.WriteLine(a[j]+", ");}
```

```
if(a[i]>a[i+1]) {  
  tmp  = a[i];  
  a[i]  = a[i+1];  
  a[i+1] = tmp;  
}
```

```
Console.WriteLine();  
for (int j=0; j<n; j++) {Console.WriteLine(a[j]+", ");}
```

Java esetén:

- `Length` helyett `length`
- `Console.Write` helyett `System.out.print`
- `Console.WriteLine` helyett `System.out.println`



Előtte: 22, 5, 4, 2,

Utána: 5, 22, 4, 2,

n-1 csere, ha $a[i] > a[i+1]$

A tömböt egyszer végigjárjuk az egymás melletti tömbelemeket kicseréljük, ha a nagyobb indexű elem értéke kisebb.

Mondatszerű leírás:

```
ciklus j=0 -tól n-2 -ig
  ha  $a[j] > a[j+1]$  akkor
    tmp = a[j]
    a[j] = a[j+1]
    a[j+1] = tmp
  ha vége
ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};
int n = a.Length; //a tömb mérete
int tmp;
```

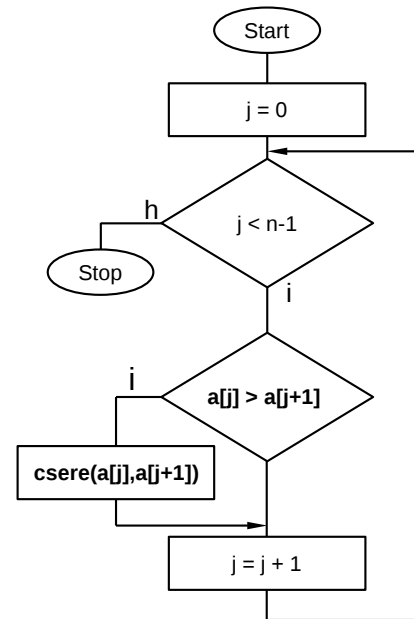
```
for (int k=0;k<n;k++)
{Console.WriteLine(a[k]+", ");}
```

```
for (int j=0;j<n-1;j++) {
  if(a[j]>a[j+1]) {
    tmp = a[j];
    a[j] = a[j+1];
    a[j+1] = tmp;
  }
}
```

```
Console.WriteLine();
for (int k=0;k<n;k++) {Console.WriteLine(a[k]+", ");}
```

Java esetén:

- Length helyett length
- Console.Write helyett System.out.print
- Console.WriteLine helyett System.out.println



Előtte: 22, 5, 4, 2,

1. csere: 22, 5, 4, 2,
2. csere: 5, 22, 4, 2,
3. csere: 5, 4, 22, 2,

Utána: 5, 4, 2, 22,

$(n-1)*(n-1)$ csere, ha $a[i] > a[i+1]$

A tömböt $n-1$ -szer végigjárjuk az egymás melletti többelemeket kicseréljük, ha a nagyobb indexű elem értéke kisebb.

A végeredmény rendezett tömb lesz, de igen sok felesleges műveletet végzünk.

Mondatszerű leírás:

```
Ciklus i := 0 -tól n-2 -ig
Ciklus j := 0-tól n-2 -ig
  Ha  $a[j] > a[j+1]$  akkor
    Csere(  $a[j]$ ,  $a[j+1]$  )
  Ciklus vége
Ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};
int n = a.Length; //a tömb mérete
int tmp;
```

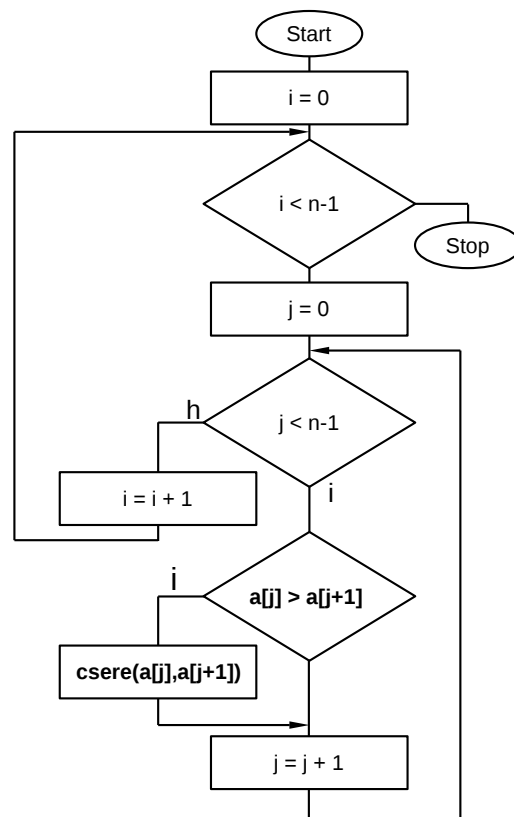
```
for (int k=0;k<n;k++)
{Console.Write(a[k]+", ");}
```

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

```
Console.WriteLine();
for (int k=0;k<n;k++) {Console.Write(a[k]+", ");}
```

Java esetén:

- Length helyett length
- Console.Write helyett System.out.print
- Console.WriteLine helyett System.out.println



Előtte: 22, 5, 4, 2,

1. külső ciklus végén: 5, 4, 2, 22,
2. külső ciklus végén: 4, 2, 5, 22,
3. külső ciklus végén: 2, 4, 5, 22,

Utána: 2, 4, 5, 22

Buborék rendezés

Az egymás utáni elemeket összehasonlítjuk. Ha a nagyobb értékű elem alacsonyabb indexű helyen van, akkor kicseréljük őket.

A tömbön először végighaladva a legnagyobb elem a legnagyobb sorszámú helyen lesz.

A külső ciklus ciklusváltozója $n-1$ -től csökken 1 -ig. A belső ciklus csak 0 -tól $i-1$ -ig tart.

Mondatszerű leírás:

```
Ciklus i := n-1 -től 1 -ig
Ciklus j := 0-től i-1 -ig
  Ha a[j] > a[j+1] akkor
    Csere( a[j], a[j+1] )
  Ciklus vége
Ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};
int n = a.Length; //a tömb mérete
int tmp;
```

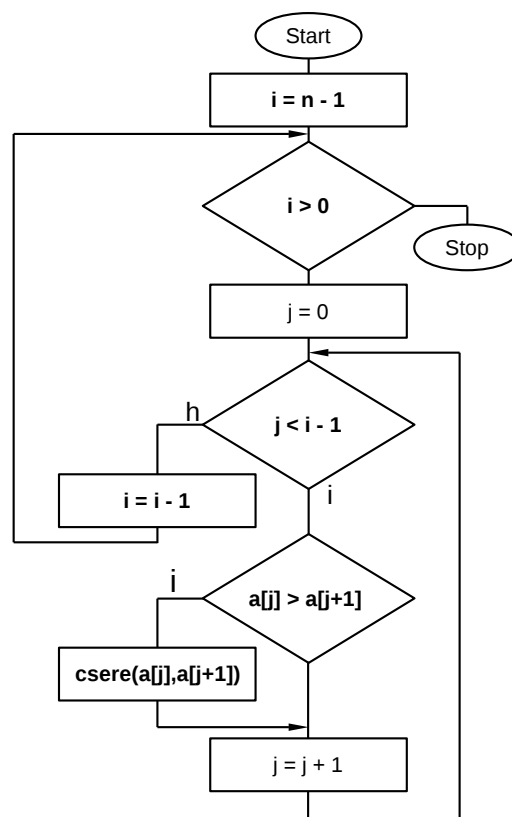
```
for (int k=0;k<n;k++)
{Console.Write(a[k]+", ");}
```

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

```
Console.WriteLine();
for (int k=0;k<n;k++) {Console.Write(a[k]+", ");}
```

Java esetén:

- `Length` helyett `length`
- `Console.Write` helyett `System.out.print`
- `Console.WriteLine` helyett `System.out.println`



Előtte: 22, 5, 4, 2,

1. külső ciklus végén: 5, 4, 2, 22,
2. külső ciklus végén: 4, 2, 5, 22,
3. külső ciklus végén: 2, 4, 5, 22,

Utána: 2, 4, 5, 22

Cserés rendezés

A tömb 1..N eleme közül kiválasztjuk a legkisebbet, majd azt a legelső elem helyére tesszük.

A tömb 2..N eleme közül kiválasztjuk a legkisebbet, majd azt a második elem helyére tesszük.

Mondatszerű leírás:

```
Ciklus i := 0 -tól n-2 -ig
  Ciklus j := i+1-től n-1 -ig
    Ha a[i] > a[j] akkor
      Csere( a[i], a[j] )
    Ciklus vége
  Ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};
int n = a.Length; //a tömb mérete
int tmp;
```

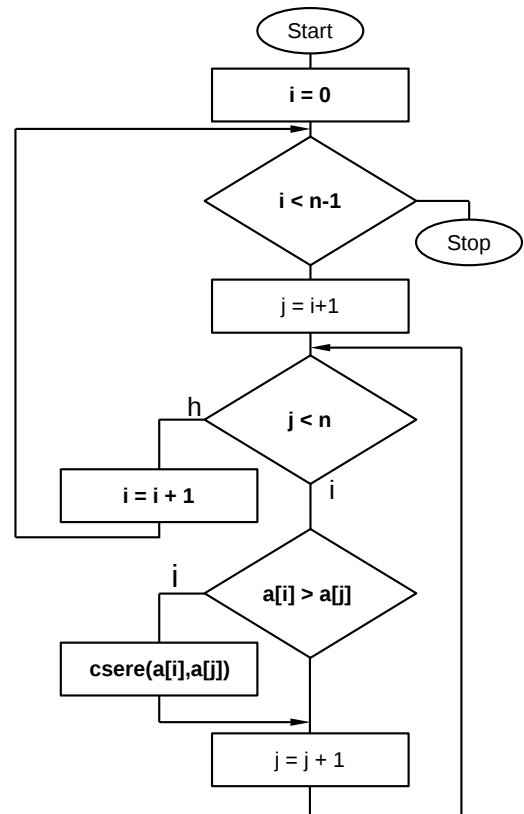
```
for (int k=0;k<n;k++)
{Console.Write(a[k]+", ");}
```

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

```
Console.WriteLine();
for (int k=0;k<n;k++) {Console.Write(a[k]+", ");}
```

Java esetén:

- `Length` helyett `length`
- `Console.Write` helyett `System.out.print`
- `Console.WriteLine` helyett `System.out.println`



Előtte: 22, 5, 4, 2,

1. külső ciklus végén: 2, 22, 5, 4,
2. külső ciklus végén: 2, 4, 22, 5,
3. külső ciklus végén: 2, 4, 5, 22,

Utána: 2, 4, 5, 22

Minimum kiválasztásos rendezés

Működése hasonló a cserés rendezéshez, a csere csak a belső ciklus után valósul meg. A belső ciklus csak értékadást tartalmaz, aminek futási ideje lényegesen rövidebb.

Mondatszerű leírás:

```
Ciklus i := 0 -tól n-2 -ig  
  min = i;  
  Ciklus j := i+1-től n-1 -ig  
    Ha a[i] > a[j] akkor  
      min = j  
  Ciklus vége  
  Csere( a[i], a[min] )  
Ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};  
int n = a.Length; //a tömb mérete  
int tmp, min;
```

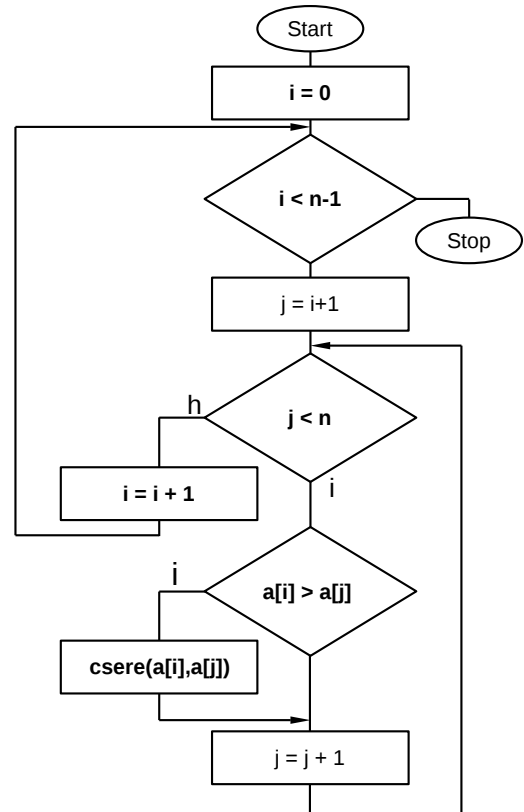
```
for (int k=0;k<n;k++)  
{Console.Write(a[k]+", ");}
```

```
for (int i=0;i<n-1;i++) {  
  min = i;  
  for (int j=i+1;j<n;j++) {  
    if(a[i]>a[j]) {  
      min = j;  
    }  
  }  
  tmp = a[min];  
  a[min] = a[i];  
  a[i] = tmp;  
}
```

```
Console.WriteLine();  
for (int k=0;k<n;k++) {Console.Write(a[k]+", ");}
```

Java esetén:

- Length helyett length
- Console.Write helyett System.out.print
- Console.WriteLine helyett System.out.println



Előtte: 22, 5, 4, 2,

1. külső ciklus végén: 2, 5, 4, 22,
2. külső ciklus végén: 2, 4, 5, 22,,
3. külső ciklus végén: 2, 4, 5, 22,

Utána: 2, 4, 5, 22

Beszúrásos rendezés

Sorra egymás után kiemeljük a tömb elemeit. Ha a kiemelt elem előtt nála nagyobb elemeket találunk, akkor azokat egygel hátrább másoljuk.

Mondatszerű leírás:

```
Ciklus i = 1 -től n-1 -ig
  kulcs = a[i]
  j = i-1
  Ciklus amíg j >= 0 és a[j]>kulcs
    a[j+1] = a[j]
    j = j - 1
  Ciklus vége
  a[j+1] = kulcs
Ciklus vége
```

C# kód

```
int[] a = {22, 5, 4, 2};
int n = a.Length; //a tömb mérete
int j, kulcs;
```

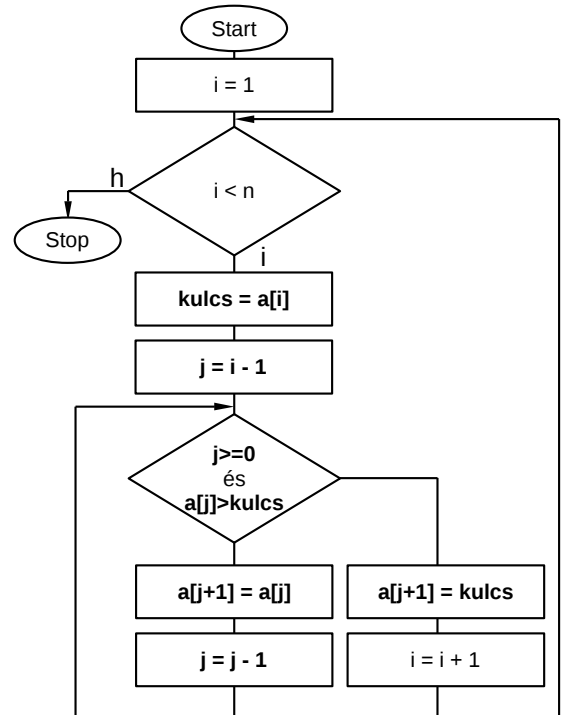
```
for (int k=0;k<n;k++)
{Console.Write(a[k]+", ");}
```

```
for (int i=1;i<n;i++) {
  kulcs = a[i];
  j = i-1;
  while(j >= 0 && a[j]>kulcs) {
    a[j+1] = a[j];
    j = j - 1;
  }
  a[j+1] = kulcs;
}
```

```
Console.WriteLine();
for (int k=0;k<n;k++)
{Console.Write(a[k]+", ");}
```

Java esetén:

- `Length` helyett `length`
- `Console.Write` helyett `System.out.print`
- `Console.WriteLine` helyett `System.out.println`



Előtte: 22, 5, 4, 2,
Kulcs=5 > 5, 22, 4, 2,
Kulcs=4 > 4, 5, 22, 2,
Kulcs=2 > 2, 4, 5, 22,
Utána: 2, 4, 5, 22,