

R02. Összefésülő rendezés (rekurzív)*

Leírás

Az algoritmus lényegét a **Rendez (E, U)** rekurzív függvény adja, ez rendezi az elemeket az elsőtől (E) az utolsóig (U).

A rendezés úgy működik, hogy meghívja önmagát a tömb első felére (E, K) és a második felére (K+1, U), majd a kapott két részben rendezett sorozatot összefésüli.

Az összefésülés úgy zajlik, hogy a két sorozat egyikéből mindig veszi a kisebbet, vagyis növekvően rendez. Az algoritmus végén lévő két ciklus már csak hozzáveszi a maradék elemeket a másik sorozatból, ha az egyik sorozat elfogyott.

Az algoritmus egy körben csak részlegesen rendezne, de mivel egyre kisebb részsorozatokra hívja meg önmagát, legalacsonyabb szinten már csak két elem marad, ezek összefésülése után folyamatosan tér vissza a nagyobb elemszámú részekre.

Algoritmus

Rendez (E, U) :

Ha $E < U$ akkor

$K := (E+U) / 2$ Rendez (E, K) ; Rendez (K+1, U)

Összefésül (E, K, U)

Elágazás vége

Eljárás vége.

Összefésül (E, K, U) :

$i := E$; $j := K+1$; $D := E-1$;

$Y := X$

Ciklus amíg $i \leq K$ és $j \leq U$

$D := D+1$

Ha $Y[i] < Y[j]$ akkor

$X[D] := Y[i]$; $i := i+1$

különben

$X[D] := Y[j]$; $j := j+1$

Elágazás vége

Ciklus vége

Ciklus amíg $i \leq K$

$D := D+1$; $X[D] := Y[i]$; $i := i+1$

Ciklus vége

Ciklus amíg $j \leq U$

$D := D+1$; $X[D] := Y[j]$; $j := j+1$

Ciklus vége

Eljárás vége.

Programkód (C#)

```
using System;
```

```
namespace Rendezés
```

```
{
```

```
    internal class Program
```

```
    {
```

* A leírás a 2017/2018-as tanév informatika OKTV 1. forduló 4. feladata alapján készült. A feladatsor javítási útmutatója elérhető [>>>itt](#)

```

//A rendezendő tömb
static int N = 20;
static int[] X = new int[N];

static void Main(string[] args)
{
    Random rnd = new Random();
    for (int i = 0; i < N; i++)
        X[i] = rnd.Next(0, 100);
    Console.WriteLine("A rendezendő tömb:");
    for (int i = 0; i < N; i++)
        Console.Write(X[i] + " ");
    Console.WriteLine();

    Rendez(0, N - 1);

    Console.WriteLine("A rendezett tömb:");
    for (int i = 0; i < N; i++)
        Console.Write(X[i] + " ");
    Console.WriteLine();
}

static void Rendez(int E, int U)
{
    if (E < U)
    {
        int K = (E + U) / 2;
        Rendez(E, K); Rendez(K + 1, U);
        Összefésül(E, K, U);
    }
}

static void Összefésül(int E, int K, int U)
{
    int i = E; int j = K + 1; int D = E - 1;
    //Segéd tömb létrehozása. Mehet elemenként is
    //Y=X nem működik, mert a tömb referenciatípus
    int[] Y = new int[N];
    Array.Copy(X, Y, N);

    //Amíg mindkét részsorozatban van elem
    //mindig vesszük a kisebbet
    while (i <= K && j <= U)
    {
        D++;
        if (Y[i] < Y[j])
            { X[D] = Y[i]; i++; }
        else
            { X[D] = Y[j]; j++; }
    }
    //Ha az egyik elfogyott, a másik maradékát a végére írjuk
    while (i <= K)
        { D++; X[D] = Y[i]; i++; }
    while (j <= U)
        { D++; X[D] = Y[j]; j++; }
}
}

```