MinimumRendez

A Maximum rendezés ennek az analógiája, csak nem a legkisebb, hanem a legnagyobb érték helyét jegyezzük meg, illetve természetesen nem a legelső, hanem a legutolsó hellyel kezdjük az elemek átrendezését /máshogy fogalmazva, az első megtalált elemet az utolsó helyre tesszük, a másodikat az utolsó előttire. Ehhez legegyszerűbb a felírási helyet meghatározó ciklust – a külső ciklust – fordítva használjuk, a tömb elem számától indul, 0-ig tart, is minden végrehajtás után eggyel csökken /i--/.

namespace MinimumRendez

{

internal class MinimumRendez

{

static void Main(string[] args)

{

// tömb létrehozása

var tomb = new int[] { 9, 6, 0, 0, 1, 2, 2, 2, 3, 1, 5, 4, 8, 2, 8, 6 };

// a legkisebb elem indexének tárolására

int MinIndex;

// változó a csere végrehajtásához

int CsereHely;

// kiírjuk az eredeti tömböt

Console.WriteLine("Rendezés előtt:");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// a legkisebb elem indexét 0-ra, az első elem indexére állítjuk

MinIndex = 0;

// a külső ciklus végig megy a tömbön

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

{

// a legkisebb elem indexét az aktuális - külső ciklus szerinti - elemszámra állítjuk

MinIndex = i;

// a belső ciklus az aktuális elemtől indul

for (int j = i; j < tomb.Length; j++)

{

// megvizsgáljuk, hogy az aktuális elem kisebb-e mint az eltárolt index által hivatkozott elem

if (tomb[j] < tomb[MinIndex])

{

// igaz ág, cseréljük a legkisebb elem indexét, mert kisebb ez az elem

MinIndex = j;

}

// megtaláltuk a legkisebb elemet, végrehajtjuk a cserét

// a legkisebb elemet a külső ciklus által mutatott helyre cseréljük /az ottani elemet meg a legkisebb helyére tesszük/

if (tomb[i] != tomb[MinIndex])

{

// a Cserehely segítségével kicseréljük a két elemet

CsereHely = tomb[i];

tomb[i] = tomb[MinIndex];

tomb[MinIndex] = CsereHely;

}

}

}

// kiírjuk a rendezett tömböt

Console.WriteLine("Minimum rendezés");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// várunk egy billentyű leütésre

Console.ReadKey();

}

}

}

Egyszerű cserés rendezés

namespace EgyszeruCsere

{

internal class EgyszeruCsere

{

static void Main(string[] args)

{

// tömb létrehozása

var tomb = new int[] { 9, 6, 0, 0, 1, 2, 2, 2, 3, 1, 5, 4, 8, 2, 8, 6 };

// változó a csere végrehajtásához

int CsereHely;

// kiírjuk az eredeti tömböt

Console.WriteLine("Rendezés előtt:");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// külső ciklus, a tömb elemeinek számával megegyező végrehajtással

for (int i = 0; i < tomb.Length-1; i++)

{

//belső ciklus a hasonlított elem kiválasztásához

for (int j = i + 1; j < tomb.Length; j++)

{

// a két egymás melletti elem összehasonlítása

if (tomb[i] > tomb[j])

{

//igaz ág, csere szükséges

CsereHely = tomb[i];

tomb[i] = tomb[j];

tomb[j] = CsereHely;

}

}

}

// kiírjuk a rendezett tömböt

Console.WriteLine("Egyszerű cserés rendezés");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// várunk egy billentyű leütésre

Console.ReadKey();

}

}

}

Buborék rendezés

namespace BuborekRendezes

{

internal class BuborekRendezes

{

static void Main(string[] args)

{

// tömb létrehozása

var tomb = new int[] { 9, 6, 0, 0, 1, 2, 2, 2, 3, 1, 5, 4, 8, 2, 8, 6 };

// változó a csere végrehajtásához

int CsereHely;

// kiírjuk az eredeti tömböt

Console.WriteLine("Rendezés előtt:");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// külső ciklus, a tömb elemszámánál eggyel kevesebb végrehajtással /a végén az utolsó helyet már nem kell rendezni/

// visszafele számol

for (int i = tomb.Length-1; i > 0; i--)

{

// belső ciklus, a hasonlítandó elemek meghatározásához

for (int j = 0; j<i; j++)

{

// összehasonlítjuk az aktuális és az eggyel mellette levő elemet

if (tomb[j] > tomb[j + 1])

{

// igaz ág, a hátsó elem /j+1/ nagyobb

// végrehajtjuk a cserét

CsereHely = tomb[j+1];

tomb[j+1] = tomb[j];

tomb[j] = CsereHely;

}

}

}

// kiírjuk a rendezett tömböt

Console.WriteLine("Buborék rendezés");

foreach (var elem in tomb)

{

Console.Write("{0},", elem);

}

Console.WriteLine();

// várunk egy billentyű leütésre

Console.ReadKey();

}

}

}