Az oszd meg és uralkodj (divide et impera) módszer

I. rész

Bevezetés, a módszer általános bemutatása

Gyakori, hogy kisebb mennyiségű adattal könnyebben lehet elvégezni valamilyen tevékenységet, sőt, ha egyetlen adatra kell elvégezni, akkor nagyon egyszerű, egy feltétel megvizsgálásával eldönthető a válasz. Az *oszd meg és uralkodj* módszer akkor alkalmazható, ha nagy mennyiségű adattal, egy nagy adathalmazzal, kell elvégezni valamilyen tevékenységet, de a feladat olyan, hogy ha ezt a nagy adathalmazt felbontjuk diszjunkt részhalmazokra, és megoldjuk azokra a feladatot, akkor a diszjunkt részhalmazokra kapott eredményből nagyon egyszerűen származtatni lehet a megoldást a teljes adathalmazra. Lényeges, hogy a feladat olyan kell legyen, hogy azt (vagy egy nagyon hasonló feladatot) kisebb adathalmazra is ki lehessen jelenteni, és ha két adathalmazra sikerült meghatározni a megoldást, akkor a két megoldás alapján egyszerűen meg lehessen adni a két halmaz egyesítéséből származó, eredeti adathalmazra a megoldást.

Általában úgy járunk el, hogy az eredeti adathalmazt két, nagyjából egyenlő számosságú (de nem feltétlenül egyenlő) részhalmazra bontjuk, ezeket ugyanúgy két-két, nagyjából egyenlő számosságú részhalmazra bontjuk és így haladunk tovább, amíg egy elemű halmazokhoz jutunk (vagy a lehető legjobban leredukáljuk a részhalmazok számosságát). Általában bármilyen tevékenység, feladat elvégzése egyetlen egyszerű adatra nagyon egyszerű. Megoldjunk az egy elemű részhalmazokra a feladatot, és ezekből az eredményekből származtatjuk a megoldását a teljes adathalmazra.

Mivel ugyanazt a feladatot oldjuk meg a részhalmazokra is, mint amit az első szétbontásnál kijelentettünk a keletkező két részhalmazra, kézenfekvő a módszert rekurzívan implementálni, de ugyanúgy iteratívan is implementálható. A kézenfekvő rekurzív implementáció miatt javaslom, hogy ez a feladatmegoldó módszer legyen az első a négy közül, amit megtanítunk, és lehetőleg azonnal a rekurzió fejezete után következzen. Azt is meg lehet csinálni, hogy a rekurzió tanítása-tanulása közben vezetjük be a módszert.

Bevezető feladat: Bináris keresés

Adott egy természetes számokat tartalmazó sorozat, amelynek elemei növekvő sorrendbe rendezettek. Határozzuk meg, hogy a sorozat tartalmaz-e egy adott természetes számot.

A feladat elemzése és megoldása

A bináris keresés módszere valószínűleg jól ismert.

Adott érték keresése egy sorozatban nem jelent nagy problémát, ez mindig megoldható szekvenciális kereséssel, de vegyük észre, hogy ebben az esetben rendezett sorozatról van szó. Vajon mennyiben könnyíti meg, gyorsítja meg a munkánkat ez a feltétel?

Tekintsük a következő sorozatot, amelynek elemei növekvő sorrendbe rendezettek:

```
1 3 4 7 9 13 18 19 21
```

Döntsük el, hogy a `13` eleme-e vagy sem a sorozatnak. Persze, hogy szekvenciális kereséssel is nagyon hamar megtaláljuk, mivel látszik, hogy a `13` a sorozat hatodik eleme, hat összehasonlítást igényelne, ha szekvenciális keresést alkalmazunk (de ha az utolsó elem lenne, akkor nyolc összehasonlítást). De gondoljunk arra, hogy a sorozat elemei növekvő sorrendbe rendezettek. Ha a keresett értéket összehasonlítjuk bármelyik elemmel, akkor nagyon könnyen eldönthetjük, hogy egyenlő vagy sem azzal (megtaláltuk) vagy a sorozatban az illető elem előtt illetve az illető elem után kell elhelyezkedjen. Erre az észrevételre alapozzuk a megoldás gondolatmenetét.

Például, ha a `13`-at összehasonlítom a sorozat középső elemével, amely az ötödik elem, vagyis a `9`. Mivel a `13` nagyobb a `9`-nél, biztos, hogy a `13` a sorozatban a `9` után kell legyen, ha a sorozat tartalmazza a `13`-at. Ezáltal az adathalmazt, amelyen dolgozok leredukáltuk a felére, vagyis a `9`-es utáni részre. Tehát a hatodik elemtől az utolsóig tartó részben keressük tovább a `13` értéket és ebben a részben is összehasonlítjuk a `13`-at a középsővel, amely (az eredeti sorozatbeli pozíciót tekintve) a hetedik elem, értéke `18`. Mivel a keresett szám kisebb a `18`-nál, ezért ha eleme a sorozatnak, akkor a `18` előtt kell elhelyezkedjen. Tehát a továbbiakban elégséges, ha a keresést a sorozat hatodik elemétől hatodik eleméig tartó részben végezzük, vagyis az adatennyiségünk egy elemre redukálódott. Mivel a hatodik elem értéke pont `13`, megtaláltuk a keresett értéket. Ugyanakkor azt is vegyük észre, hogy meghatároztuk a keresett elem pozícióját (hatodik) a sorozatban. Az is feltűnő, hogy ha szekvenciális keresést hajtunk végre, akkor hat összehasonlítás szükséges ahhoz, hogy megtaláljuk a `13`-at, de bináris keresést alkalmazva elég volt három összehasonlítás.

Vigyük végig a gondolatmenetet. Szükségünk lesz két változóra, amelyek tárolják, hogy a sorozat hányadik elemétől, hányadik eleméig tartó részben végezzük a keresést. Nevezzük ezeket a változókat `bal` és `jobb`-nak. Kezdetben az egész sorozattal dolgozunk, tehát a `bal` értéke `1`, a `jobb` értéke a sorozat elemeinek száma lesz és ahogy folyamatosan haladunk, a `bal` értéke egyre nő, a `jobb` értéke pedig csökken. A sorozat elemeit is tárolni kell, amire használjuk a `T` tömböt. A példában szögletes zárójelekkel is jelzem, hogy a sorozat melyik részében tartunk a keresés során.

`bal`	`1`	`jobb`	`9`
`T`	`[1 3 4 7 9 13 18 19 21]`		

Meg kell határozzuk, hogy hol helyezkedik el a középső elem, mert leghamarabb ezzel kell összehasonlítsuk. Egyértelmű, hogy ennek a pozícióját a `bal` és `jobb` értékeinek számtani középarányosa adja. Ennek az adatnak a tárolása érdekében vezessük be a `kozep` nevű változót. Tehát `kozep ← [(bal + jobb) / 2]`, ami `5`. A példánál kövessük folyamatosan a `kozep` változó értékét is.

`bal`	`1`	`jobb`	`9`	`kozep`	`5`
`T`	`[1 3 4 7` **9** `13 18 19 21]`				

Hasonlítsuk össze a keresett értéket, esetünkben a `13`-at, a középső (ötödik) elemmel, amelynek értéke `9`. Mivel a keresett érték nagyobb a `9`-nél és a sorozat növekvő sorrendbe rendezett, ezért a keresett érték mindenképp a `9` után kell legyen, ha a sorozat tartalmazza a `13`-at. Ennek következtében elég a keresést a hetedik elem utáni részben végezni, vagyis `bal ← kozep + 1`. Így az eredeti adathalmazunk számossága a felére csökkent, és a sorozat jobb oldalára kerültünk a kereséssel, átugorva az összes bal oldalon levő elemet.

bal	6	jobb	9	kozep	5
T	1 3 4 7 **9** [13 18 19 21]				

Most a hatodik elemtől a kilencedik elemig tartó részben vagyunk. Ebben a részben a középsővel szeretnénk összehasonlítani. A `kozep` változó számolására megadott képlet alapján:

bal	6	jobb	9	kozep	7
T	1 3 4 7 9 [13 **18** 19 21]				

A keresett érték kisebb mint `18`, ezért a `18` előtt kell elhelyezkedjen. A részt amelyben a keresést végezzük megint csökkentjük: `jobb ← kozep - 1`.

bal	6	jobb	6	kozep	7
T	1 3 4 7 9 [13] 18 19 21				

Most jutottunk el az egy elemű részhez. Ebben a részben a középső is maga az az egy elem lesz.

bal	6	jobb	6	kozep	6
T	1 3 4 7 9 [**13**] 18 19 21				

És ha ezzel összehasonlítjuk a keresett értéket, akkor elmondhatjuk, hogy megtaláltuk, méghozzá a hatodik pozícióban (amit a `kozep` értéke ad).

Most lássuk mi történik, ha egy olyan értéket keresünk, amely nincs benne a sorozatban. Próbáljuk szintén az előző sorozatban megkeresni a `8`-as értéket. Ugyanúgy indulunk, vagyis:

bal	1	jobb	9	kozep	5
T	[1 3 4 7 **9** 13 18 19 21]				

A `8` kisebb a `9`-nél, tehát:

bal	1	jobb	4	kozep	2
T	[1 **3** 4 7] 9 13 18 19 21				

Most a `3`-mal hasonlítjuk össze a `8`-at, mivel nagyobb annál:

bal	3	jobb	4	kozep	3
T	1 3 [**4** 7] 9 13 18 19 21				

Most a `4`-gyel hasonlítjuk össze a `8-at`, nagyobb annál:

bal	4	jobb	4	kozep	4
T	1 3 4 [7] 9 13 18 19 21				

Eljutottunk az egy elemű részhez. A 7-tel hasonlítjuk össze a 8-at, nagyobb annál:

bal	5	jobb	4	kozep	
T	1 3 4 7] [9 13 18 19 21				

Lám, ha a keresett érték nincs benne a sorozatban, akkor előbb-utóbb olyan helyzetbe kerülünk, hogy a bal értéke meghaladja a jobbét, vagyis az a rész amiben a keresést végezzük üres halmaz lesz. Ez lesz a feltétele annak, hogy leállíthassuk a keresést akkor is, ha a keresett elem nincs benne a sorozatban.

Mivel a keresett értéket mindig a sorozatrészünk középső elemével hasonlítjuk össze, vagy úgy fejeződik be a keresés, hogy egy ilyen középsőként megtaláljuk a keresett értéket (gondoljunk csak arra, ha pont a 9 értéket keressük, akkor az első összehasonlítással megtaláljuk) vagy pedig úgy fejeződik be, hogy a rész amiben keresünk, üressé válik, vagyis bal > jobb.

Mindezek ismeretében most már egyszerű megírni az algoritmust. Megírjuk mind rekurzívan, mind iteratívan.

Az algoritmus pszeudokódban iteratívan

Az algoritmusunk eredménye a keresett érték pozíciója, ha megtaláljuk azt a sorozatban, különben az eredmény -1.

```
Algoritmus BinárisKeresés
Adottak:    n,              {a sorozat elemeinek száma}
            Ti (i=1,n),          {a rendezett sorozat elemei}
            k                    {a keresett érték}
bal ← 1
jobb ← n
Ismételd
    kozep ← [(bal + jobb) / 2]
    Ha k < Tkozep akkor      {bal oldali részben kell keresni}
        jobb ← kozep - 1
    kulonben            {jobb oldali részben kell keresni}
        bal ← kozep + 1
    (Ha) vége
Ameddig bal > jobb VAGY Tkozep = k
Ha Tkozep = k akkor
    er ← kozep
különben
    er ← -1
(Ha) vége
Eredmény: er
Algoritmus vége
```

Az algoritmus pszeudokódban rekurzívan

Egy rekurzív függvényt készítünk, amely eredményként adja a meg a keresett érték pozícióját, ha megtaláljuk azt a sorozatban, és különben az eredménye legyen `-1`.

```
Függvény keres(k, bal, jobb, T)
    Ha bal > job akkor              {ha üres a rész, nem találtuk meg}
        er ← -1
    különben
        kozep ← [(bal + jobb) / 2]
        Ha Tkozep = k akkor         {ha egyenlő a középsővel, megtaláltuk}
            er ← kozep
        különben
            Ha k < Tkozep akkor     {bal oldali részben kell keresni}
                er ← keres(k, bal, kozep - 1, T)
            kulonben                {jobb oldali részben kell keresni}
                er ← keres(k, kozep + 1, jobb, T)
            (Ha) vége
        (Ha) vége
    (Ha) vége
    Eredmény: er
Függvény vége
```

Algoritmus BinárisKeresés

```
Adottak:    n,              {a sorozat elemeinek száma}
            Ti (i=1,n),     {a rendezett sorozat elemei}
            k               {a keresett érték}
    er ← keres(k, 1, n, T)
Eredmény: er
Algoritmus vége
```

Szemléltető feladat: Sorozat legnagyobb eleme

Adott egy természetes számokat tartalmazó sorozat. Határozzuk meg a legnagyobb elemét oszd meg és uralkodj módszert használva.

A feladat elemzése és megoldása

Az alapötlet az, hogy a közepénél két felé választjuk a sorozatot. Ha meghatározzuk a bal oldali rész legnagyobb elemét, majd meghatározzuk a jobb oldali rész legnagyobb elemét is, akkor a két elem közül a nagyobb lesz a teljes sorozat legnagyobb eleme. A bal és jobb oldali részek legnagyobb elemeinek meghatározása céljából is rekurzívan ugyanezt a kettéosztásos stratégiát alkalmazzuk. Amikor egy elemű részekhez jutottunk, akkor egyszerű meghatározni a rész legnagyobb elemét, mert az nem más, mint az az egy elem.

Lássunk egy példát. Legyenek a sorozat elemei a következők:

`2 7 4 1 9 3`

A példánál végig nyomon követjük, hogy a sorozat melyik részében vagyunk, amelyet a `bal` és `jobb` nevű változóink fognak határolni.

bal	1	jobb	6

2	7	4	1	9	3

Kettéosztjuk a sorozatot a közepénél, és rátérünk, hogy meghatározzuk a bal oldali rész legnagyobb elemét.

bal	1	jobb	3

2	7	4	1	9	3

Ezt a részt is kettéosztjuk a közepénél, és rátérünk, hogy meghatározzuk a bal oldali részének legnagyobb elemét.

bal	1	jobb	2

2	7	4	1	9	3

Mivel ez a rész sem egy elemű, ezt is kettéosztjuk, és rátérünk a bal oldali részének legnagyobb elemét meghatározni.

bal	1	jobb	1

2	7	4	1	9	3

Vegyük észre, hogy most egy egyelemű részhez jutottunk, aminek a legnagyobb eleme maga az az egy elem. Jelöljük $max_{i,j}$-vel a sorozat `i`-edik elemétől `j`-edik eleméig tartó rész legnagyobb elemét. Ennek megfelelően:

bal	1	jobb	1

2	7	4	1	9	3
$max_{1,1}$ = 2					

Most már az elsőtől a másodikig tartó rész bal oldali részének megvan a legnagyobb eleme. Meg kell határozzuk a jobb oldali rész legnagyobb elemélt is, vagyis megyünk az elsőtől a másodikig tartó rész jobb oldalára.

bal	2	jobb	2

2	7	4	1	9	3
$max_{1,1}$ = 2					

Ez is egy egyelemű rész. Legnagyobb elem maga az az egy elem, vagyis 7.

bal 2	jobb 2				
2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$				

Ebben a pillanatban sikerült meghatározni az elsőtől másodikig tartó rész bal oldali és jobb oldali részének is a legnagyobb elemét, amelyek közül a nagyobb fogja megadni az elsőtől másodikig tartó rész legnagyobb elemét. Vagyis $max_{1,2} = max\ \{max_{1,1}, max_{2,2}\} = 7$.

bal 1	jobb 2				
2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$				
$max_{1,2} = 7$					

Ebben a pillanatban sikerült meghatározni az elsőtől harmadikig tartó rész bal oldali részének legnagyobb elemét. Következik, hogy meghatározzuk a jobb oldali részének legnagyobb elemét, de az egyelemű.

bal 3	jobb 3				
2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$			
$max_{1,2} = 7$					

Ebben a pillanatban sikerült meghatározni az elsőtől harmadikig tartó rész bal oldali és jobb oldali részeinek legnagyobb elemeit. Az elsőtől harmadikig tartó rész legnagyobb eleme a bal és a jobb oldali részek legnagyobb elemei közül a nagyobb, vagyis

$max_{1,3} = max\ \{max_{1,2}, max_{3,3}\} = 7$.

bal 1	jobb 3				
2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$			
$max_{1,2} = 7$					
$max_{1,3} = 7$					

Ezzel megvan az eredeti sorozat bal oldali részének legnagyobb eleme. Hasonlóan fogjuk meghatározni a jobb oldali rész legnagyobb elemét is. A jobb oldali, negyediktől hatodikig tartó részt is kétfelé kell osztani.

bal	4	jobb	6

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$			
$max_{1,2} = 7$					
$max_{1,3} = 7$					

A középső elem az 5-ödik lesz, és rátérünk ennek a résznek a bal oldalára.

bal	4	jobb	5

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$			
$max_{1,2} = 7$					
$max_{1,3} = 7$					

Ez a rész kételemű tehát ezt is kétfelé osztjuk.

bal	4	jobb	4

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$			
$max_{1,2} = 7$					
$max_{1,3} = 7$					

Az így keletkező bal oldali rész egyelemű.

bal	4	jobb	4

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$	$max_{4,4} = 1$		
$max_{1,2} = 7$					
$max_{1,3} = 7$					

Megvan a negyediktől ötödikig tartó rész bal oldalának legnagyobb eleme. Rátérünk ennek a résznek a jobb oldalára. Ennek a résznek a jobb oldala is egyelemű, tehát könnyű meghatározni a legnagyobb elemét.

bal	5	jobb	5

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$	$max_{4,4} = 1$	$max_{5,5} = 9$	
$max_{1,2} = 7$					
$max_{1,3} = 7$					

Megvan a negyediktől ötödikig tartó rész bal oldalának és jobb oldalának is a legnagyobb eleme. A negyediktől ötödikig tartó rész legnagyobb eleme a két rész legnagyobb elemei közül a nagyobb, vagyis $max_{4,5} = max\ \{max_{4,4},\ max_{5,5}\} = 9$.

bal	4	jobb	5

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$	$max_{4,4} = 1$	$max_{5,5} = 9$	
$max_{1,2} = 7$			$max_{4,5} = 9$		
$max_{1,3} = 7$					

Ezzel a teljes jobb oldali rész bal oldalát elintéztük. Most következik a teljes jobb oldali rész jobb oldala. Ez is egy elemű.

bal	6	jobb	6

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$	$max_{4,4} = 1$	$max_{5,5} = 9$	$max_{6,6} = 3$
$max_{1,2} = 7$			$max_{4,5} = 9$		
$max_{1,3} = 7$					

És sikerült a teljes jobb oldali rész, vagyis a negyediktől hatodikig tartó rész bal és jobb oldalának is meghatározni a legnagyobb elemeit, amelyek közül a nagyobb adja meg a teljes jobboldali rész legnagyobb elemét, vagyis $max_{4,6} = max\ \{max_{4,5},\ max_{6,6}\} = 9$.

bal	4	jobb	6

2	7	4	1	9	3
$max_{1,1} = 2$	$max_{2,2} = 7$	$max_{3,3} = 4$	$max_{4,4} = 1$	$max_{5,5} = 9$	$max_{6,6} = 3$
$max_{1,2} = 7$			$max_{4,5} = 9$		
$max_{1,3} = 7$			$max_{4,6} = 9$		

És eljutottunk a végéhez. Megvan a teljes sorozat bal oldalának és jobb oldalának is a legnagyobb eleme. Ezek közül a nagyobb lesz a teljes sorozat legnagyobb eleme. Vagyis $max_{1,6} = max\ \{max_{1,3},\ max_{4,6}\} = 9$.

```
bal    4    jobb    6

    2           7            4            1            9            3
max1,1 = 2  max2,2 = 7                max4,4 = 1  max5,5 = 9
                         max3,3 = 4                             max6,6 = 3
      max1,2 = 7                            max4,5 = 9
             max1,3 = 7                          max4,6 = 9
                           max1,6 = 9
```

Az algoritmus pszeudokódban

Készítünk egy rekurzív függvényt, amely ha egyelemű résszel dolgozik, akkor visszatéríti azt az egy elemet, ha nem, akkor kétfelé osztja a részt, meghívja önmagát annak a bal és jobb oldalára. A két oldalról kapott értékek közül pedig a nagyobbat adja eredményként.

```
Függvény max(bal, jobb, T)
    Ha bal = jobb akkor                         {ha egyelemű a rész}
        er ← T_bal
    különben
        közép = [(bal + job) / 2]          {a rész közepe}
        maxbal = max(bal, kozep, T)            {max bal oldal}
        maxjobb = max(kozep + 1, jobb, T) {max jobb oldal}
        Ha maxbal > maxjobb akkor
            er ← maxbal
        különben
            er ← maxjobb
        (Ha) vége
    (Ha) vége
    Eredmény: er
Függvény vége
```

```
Algoritmus LegnagyobbElem
Adottak: n,              {a sorozat elemeinek száma}
    T_i (i=1,n),         {a rendezett sorozat elemei}
    er ← max(1, n, T)
Eredmény: er
Algoritmus vége
```

Közös vonások és általános gondolatmenet

Gyakori, hogy egy feladat lebontható olyan részfeladatokra, melyek megegyeznek vagy nagyon hasonlók az eredeti feladathoz, de kisebb mennyiségű adattal kell dolgozni a megoldásuk során. Az is tény, hogy bizonyos esetekben a kisebb mennyiségű adat általában megkönnyíti a feladat megoldását. Az oszd meg ér uralkodj (divide et impera) módszer elnevezése is onnan származik, hogy az a stratégiánk, hogy szétosszuk az adatokat és ezáltal könnyebben megoldható, az eredetivel azonos vagy ahhoz nagyon hasonló részfeladatokhoz jussunk. Ezek a részfeladatok is olyanok kell legyenek, hogy szintén az adathalmazukat megosztva részfeladatokra lehessen bontani őket. Itt máris kezd érződni a rekurzív gondolkodásmód. Ha továbbvisszük a részfeladatokra bontást, ami elsősorban az adathalmaz megosztására épül, előbb-utóbb olyan részfeladatokhoz jutunk, amelyek megoldása nagyon egyszerű. Ezeket a feladatokat nem bontjuk tovább, hanem megoldjuk és kezdjük származtatni a megoldásaikból azoknak a feladatoknak a megoldását, amelyeknek ők a részfeladataik. Általában ez az egyszerű feladat, amelyet végül is meg kell oldani úgy néz ki, hogy az adathalmazunk egyelemű halmazra redukálódott és legfeljebb egy egyszerű feltétel vizsgálatával vagy más egyszerű tevékenység elvégzésével megoldható. Az ilyen feladatot nevezzük az oszd meg és uralkodj módszer esetében triviális feladatnak.

Az oszd meg és uralkodj módszerrel megoldható feladatok közös vonásai

1. A feladat vele azonos, vagy hozzá nagyon hasonló, egymástól függetlenül megoldható részfeladatokra bontható.

Az első szemléltető feladat esetében a részfeladat ugyanaz, vagyis meghatározni, hogy egy adott érték eleme-e egy rendezett sorozatnak, de a részfeladatbeli sorozat az eredeti sorozat fele (bal vagy jobb oldala).

A második szemléltető feladat esetében is a részfeladat ugyanaz, mint az eredeti. Meghatározni egy „sorozat” (ami az eredeti sorozat egy részsorozata) legnagyobb elemét.

2. A részfeladatra bontás során el kell jussunk a triviális feladathoz, amely megoldása nagyon egyszerű.

Az első szemléltető feladat esetében kétféle triviális feladatunk létezik. Az egyik, ha megtaláljuk a keresett elemet, amikor egy sorozat(rész) középső elemével egyenlő a keresett érték. A másik triviális feladat, amikor üres sorozattal kell dolgozni, mert akkor biztos nem eleme a sorozatnak a keresett érték.

A második szemléltető feladat esetében a triviális feladat az egy elemű sorozat legnagyobb elemének meghatározása.

3. A részfeladatokból felépíthető a feladat megoldása.

Az első szemléltető feladat esetében a teljes feladat megoldása megegyezik a részfeladat megoldásával.

A második szemléltető feladat esetében két részfeladatból származó legnagyobb elemek közül a nagyobb adja a feladat megoldását.

Az oszd meg és uralkodj módszer általános gondolatmenete

1. A feladat vele azonos, vagy hozzá nagyon hasonló, olyan részfeladatokra bontása, amelyek egymástól függetlenül megoldhatók.
2. A triviális feladat meghatározása és megoldása.
3. A triviális részfeladatok megoldásaiból kiindulva a többi részfeladat és végül az eredeti feladat megoldásának megadása.

Demeter Hunor

Csodaszép, gyógyító, mérgező növényeink

Őszi kikerics

Az **őszi kikerics** *(Colchicum autumnale)* a kikericsfélék *(Colchicaceae)* családjába tartozó, augusztus-szeptemberben nyíló, évelő mérgező növény. Tudományos megnevezése a Kolkhisz ókori királyság nevéből származik. A monda szerint Médeia, a legendás kolkhiszi királylány ezt a virágot használta varázsszereinek előállításához. Mérgező hatásához kapcsolódnak a népi elnevezések is: kökörcsiny, kutyadöglesztő.

Legelőkön, erdei tisztásokon nő, gumója hártyás, tömör, fehéres színű. Lándzsa alakú levelei áprilisban kezdenek fejlődni, kifejlődve 20–30 cm hosszúak és zöld színűek. Érdekes, hogy bár levelei már korán tavasszal megjelennek, lila virágai csak ősszel nyílnak. Virága kb. 7 cm átmérőjű, 6 szirmú, jellegzetes világos lila szinű.

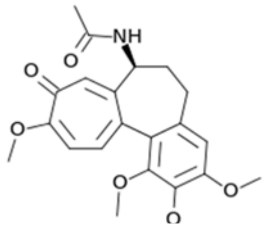
Az őszi kikerics mellett ismeretes még a homok kikerics (Colchicum arenarium), valamint a számunkra fontos **magyar kikerics**. A magyar kikericset Janka Viktor botanikus fedezte fel 1867 tavaszán a Szársomlyó hegy déli oldalán. A magyar kikerics védett növény, ez volt az első hatóságilag védett növény Magyarországon. Védettségét 1944-ben hirdették ki. A magyar kikerics képét a ma már forgalomból kivont kétforintos érme hátlapján napjainkban a védett eredetű villányi borok címkéjén láthatjuk.

Az őszi kikerics gyönyörű, vadon élő növény de szépsége miatt számos kertben is megtalálhatjuk.

Fontos tudni, hogy szépsége mellett minden része, levele, virága, gumója mérgező. Legelőinken az állatok kikerülik, nem legelik le. Kivételt képeznek a kecskék és a juhok, rájuk a kikerics mérgező anyaga nem hat, de a mérgező hatóanyag átjut a tejbe és ha az ember ilyen tejet iszik, az is károsan hat. Mérgező hatását a növényben jelenlevő kolhicin alkaloida okozza, mely főleg a növényi olajokat tartalmazó magjában található. Orvosi felhasználása főleg a köszvény kezelésére irányul, mint alkalmazható gyógyszer az USA-ban engedélyezett, kizárólagosan orvosi felügyelet mellett. Toxikus hatása miatt veszélyes, ezért csak erős köszvényes roham esetében alkalmazzák. Erős mérgező hatása a sejtosztódás gátlásán alapszik. A növényi kivonatból előállított kolhicint a növénytermesztésben alkalmazzák, mint sejtméret növelő, tömegnövelő hatóanyag. Mutagén hatása van, amit a növények nemesítésénél alkalmaznak. Napjainkban folynak kutatások a kikerics hatóanyagának felhasználására, olyan gyógyszerhatóanyag izolálására, melyet a daganatos megbetegedésekben alkalmazhatnak. Mérgező hatása miatt komoly mellékhatásokat okoz.

A mérgezés jelei : hasmenés, hányás, légszomj, súlyos esetekben keringés és légzés leállás.

A tiszta hatóanyagot először az őszi kikericsből izolálta egy francia vegyész, S. P Pelletier 1820-ban. Szerkezetét csak később, 1945-ben Michael Dewar igazolta. Szerkezete 3 különböző szerkezetű gyűrűs rendszeren alapszik, szerkezetében az alkaloidokra jellemző nitrogén az oldalláncban található számos metoxi szubsztituens mellett :



N-[(7S)-1,2,3,10-Tetrametoxy-9-oxo-5,6,7,9-tetrahydrobenzo[a]heptalen-7-il]acetamid.

A gyönyörű lila virága költőinket is megihlette :

Arany János valószínűleg az őszi kikericsről nevezte el öregkori versciklusát az *Őszikéket:*

Olvasó, ha fennakadsz, hogy
Könyvem címe „Őszikék",
Tudd meg: e néven virágok
Vannak ősszel, és - csibék.

Arany János kései versciklusának fenti, címadó költeménye tette közismertté őszike szavunkat, mely az őszi virággal együtt az őszi kikerics (*Colchicum autumnale*) Székelyföldön elterjedt megnevezése.

Guillaume Appollinaire: *Kikericsek* című versében így mutatja be az őszi kikericset:

Most mérget hajt a rét s virágzik késő őszig
Legelget a tehén
S lassan megmérgeződik
Kikericsek virítnak kékek és lilak….
(Radnóti Miklós fordítása)

Kányádi Sándor: *Őszi réten*

Őszi réten
lila-fehéren
csak viríts, csak viríts,
menyasszony-ruhás kikerics.

Ismerjük meg, gyönyörködjünk az őszi réteken virágzó kikericsben, de ne felejtsük el, hogy levelei, virága és gumója mérgező!

Majdik Kornélia