# Programozás
# 11. előadás

# Tartalom



# Rendezési feladat

## Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{H}^N$
  $\leq : \mathbb{H} \times \mathbb{H} \to \mathbb{L}$
- Kimenet: $Y_{1..N} \in \mathbb{H}^N$
- Előfeltétel: Rendezés($\leq$) és RendezettE$_{\leq}$($\mathbb{H}$)
- Utófeltétel: RendezettE$_{\leq}$(Y) és $Y \in$ Permutáció(X)
- Jelölések:
  - RendezettE$_{\leq}$(X/$\mathbb{H}$): X/$\mathbb{H}$ rendezett-e a $\leq$-ra?
  - $Y \in$ Permutáció(X): Y az X elemeinek egy permutációja-e?

# Rendezési feladat

A rendezések egy részében olyan megvalósítást választunk, amiben a bemenetnek és a kimenetnek **ugyanaz** a sorozat felel meg, azaz **helyben** rendezünk.

## Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{H}^N$, $\le : \mathbb{H} \times \mathbb{H} \rightarrow L$
- Kimenet: $X'_{1..N} \in \mathbb{H}^N$
- Előfeltétel: Rendezés($\le$) és RendezettE$_{\le}$($\mathbb{H}$)
- Utófeltétel: RendezettE$_{\le}$(X') és X'$\in$Permutáció(X)
- Jelölések:
  - X': az X kimeneti (megálláskori) értéke
  - RendezettE$_{\le}$(X/$\mathbb{H}$): X/$\mathbb{H}$ rendezett-e a $\le$-ra?
  - X'$\in$Permutáció(X): X' az X elemeinek egy permutációja-e?

# Rendezések
## (fontos új fogalmak, jelölések)

- **Aposztróf** a specifikációban:
  Ha egy adat előfordul a bemeneten és kimeneten is, akkor az UF-ben együtt kell előfordulnia az adat bemenetkori és kimenetkori értéke. Megkülönböztetésül a kimeneti értéket „megaposztrofáljuk”.
  Pl.: Z’:=a Z kimeneti (megálláskori) értéke.
- A $\leq$ reláció **rendezés**, ha
  1. *reflexív*: $\forall h \in \mathbb{H}: h \leq h$
  2. *antiszimmetrikus*: $\forall h,i \in \mathbb{H}: h \leq i$ és $i \leq h \rightarrow h = i$
  3. *tranzitív*: $\forall h,i,j \in \mathbb{H}: h \leq i$ és $i \leq j \rightarrow h \leq j$

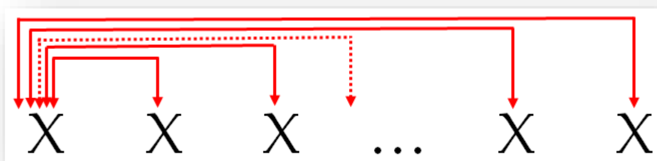
# Rendezések
## (fontos új fogalmak, jelölések)

- $\mathbb{H}$ (teljesen) **rendezett halmaz**:
  RendezettE($\mathbb{H}$):=$\forall$h,i$\in\mathbb{H}$: h$\leq$i vagy i$\leq$h
- **Rendezett sorozat**:
  RendezettE(Z):=$\forall$i(1$\leq$i$\leq$N–1): $Z_i \leq Z_{i+1}$
- **Permutáció**halmaz:
  Permutáció(Z):=a Z$\in\mathbb{H}^N$ sorozat elemeinek *összes permutáció*ját tartalmazó *halmaz*; amelynek tehát egyik eleme a kívánt rendezettségű sorozat…

# Egyszerű cserés rendezés

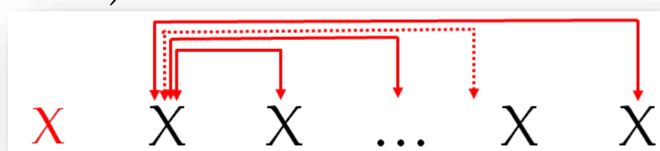
## A lényeg:

- Hasonlítsuk az első elemet az összes mögötte levővel, s ha kell, cseréljük meg!




*A minimum az „alsó” végére kerül.*

- Ezután ugyanezt csináljuk a második elemre!




- …

*A pirossal jelöltek már a helyükön vannak*

- Végül az utolsó két elemre!


# Egyszerű cserés rendezés

## Algoritmus:

**Változó**
i,j:**Egész**
s:TH

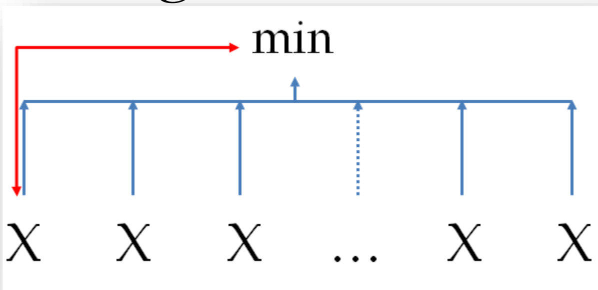
i=1..N–1
j=i+1..N
X[i]>X[j]
I | N
Elem-csere
s:=X[i]
X[i]:=X[j]
X[j]:=s
—

- Hasonlítások száma: 1+2+...+N–1= $N \cdot \frac{N-1}{2}$
- Mozgatások száma: 0 .. $3 \cdot N \cdot \frac{N-1}{2}$

# Minimum-kiválasztásos rendezés

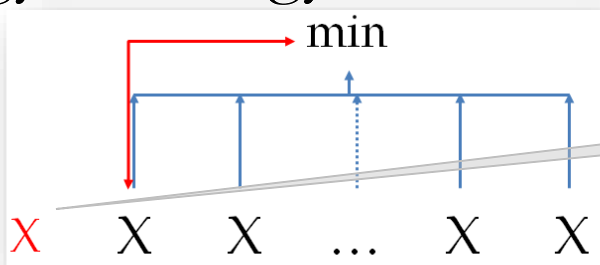
## A lényeg:

- Határozzuk meg az 1..N elemek minimumát, s cseréljük meg az 1.-vel!



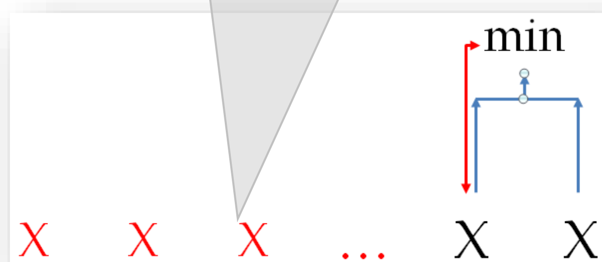

*A minimum az „alsó” végére kerül.*

- Ezután ugyanezt tegyük a 2..N elemre!




*A pirossal jelöltek már a helyükön vannak*

- …
- Végül az utolsó két (N–1..N) elemre!




# Minimum-kiválasztásos rendezés

## Algoritmus:

Minimum-kiválasztás az i.-től

Elem-csere

```
i=1..N–1
    MinI:=i
    j=i+1..N
        X[MinI]>X[j]
        I: MinI:=j
        N: —
    s:=X[i]
    X[i]:=X[MinI]
    X[MinI]:=s
```

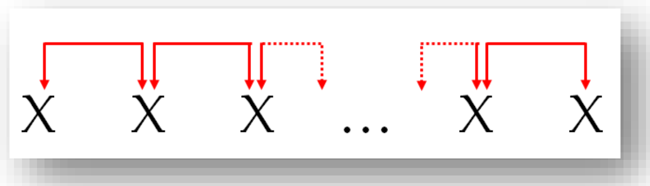
**Változó**
MinI,
i,j:**Egész**
s:TH

- Hasonlítások száma: $1+2+...+N-1=N \cdot \frac{N-1}{2}$
- Mozgatások száma: $3 \cdot (N-1)$

# Buborékos rendezés

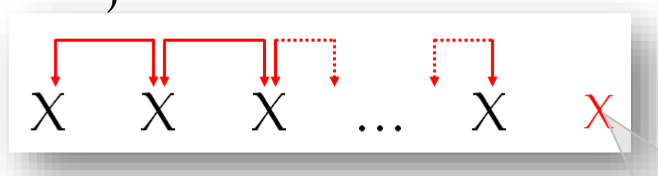
## A lényeg:

- Hasonlítsunk minden elemet a mögötte levővel, s ha kell, cseréljük meg!




*A maximum a „felső” végére kerül.*

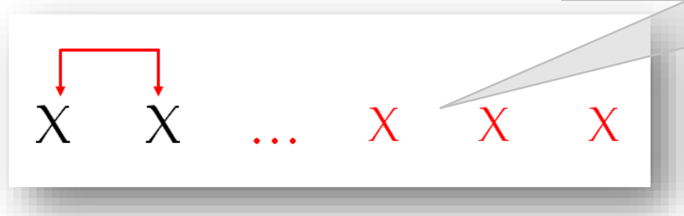
- Ezután ugyanezt csináljuk az utolsó elem nélkül!




*A többiek is tartanak a helyük felé.*

- …
- Végül az első két elemre!

*A pirossal jelöltek már a helyükön vannak*




# Buborékos rendezés

## Algoritmus:

**Változó**
i,j:**Egész**
s:TH

i=N..2, -1-esével
j=1..i–1
X[j]>X[j+1]
I
s:=X[j]
X[j]:=X[j+1]
X[j+1]:=s
N
—

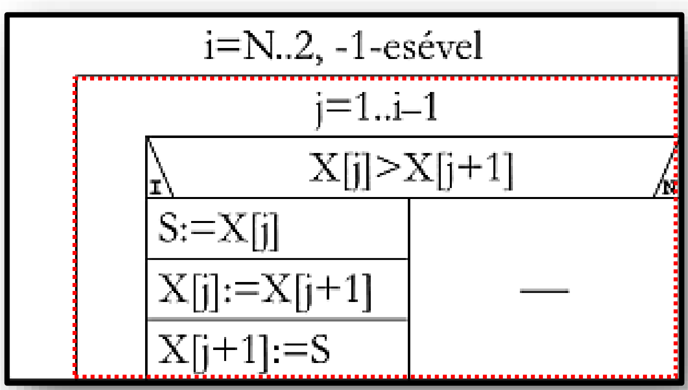
Elem-csere

- Hasonlítások száma: 1+2+...+N–1= $N \cdot \frac{N-1}{2}$
- Mozgatások száma: 0 .. $3 \cdot N \cdot \frac{N-1}{2}$

# Javított buborékos rendezés

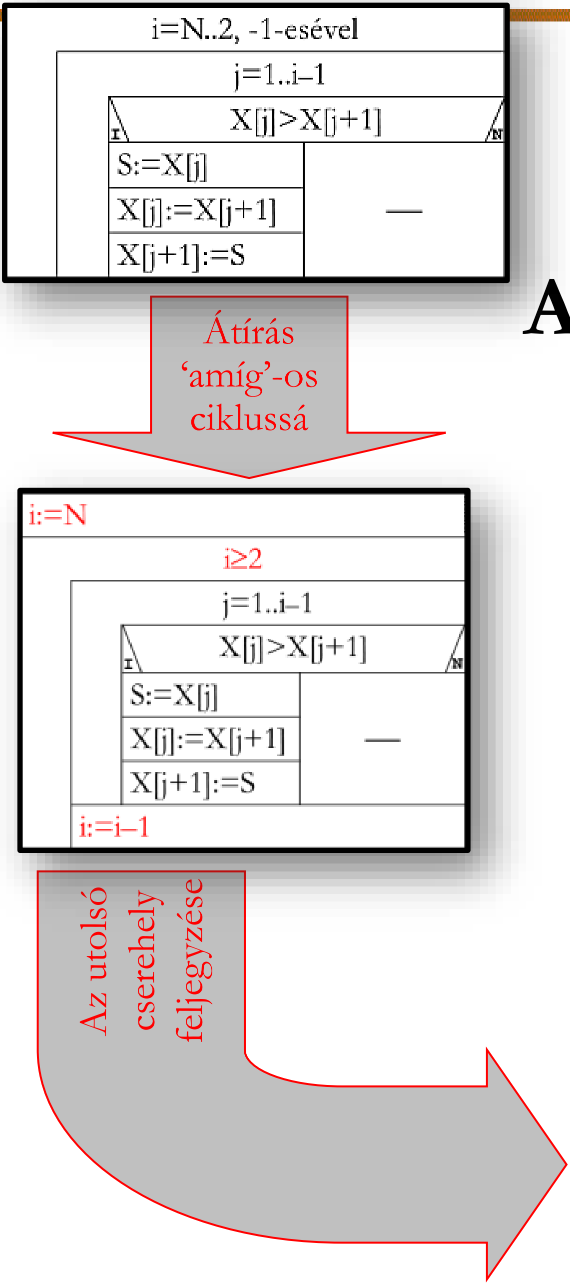
## Megfigyelések:

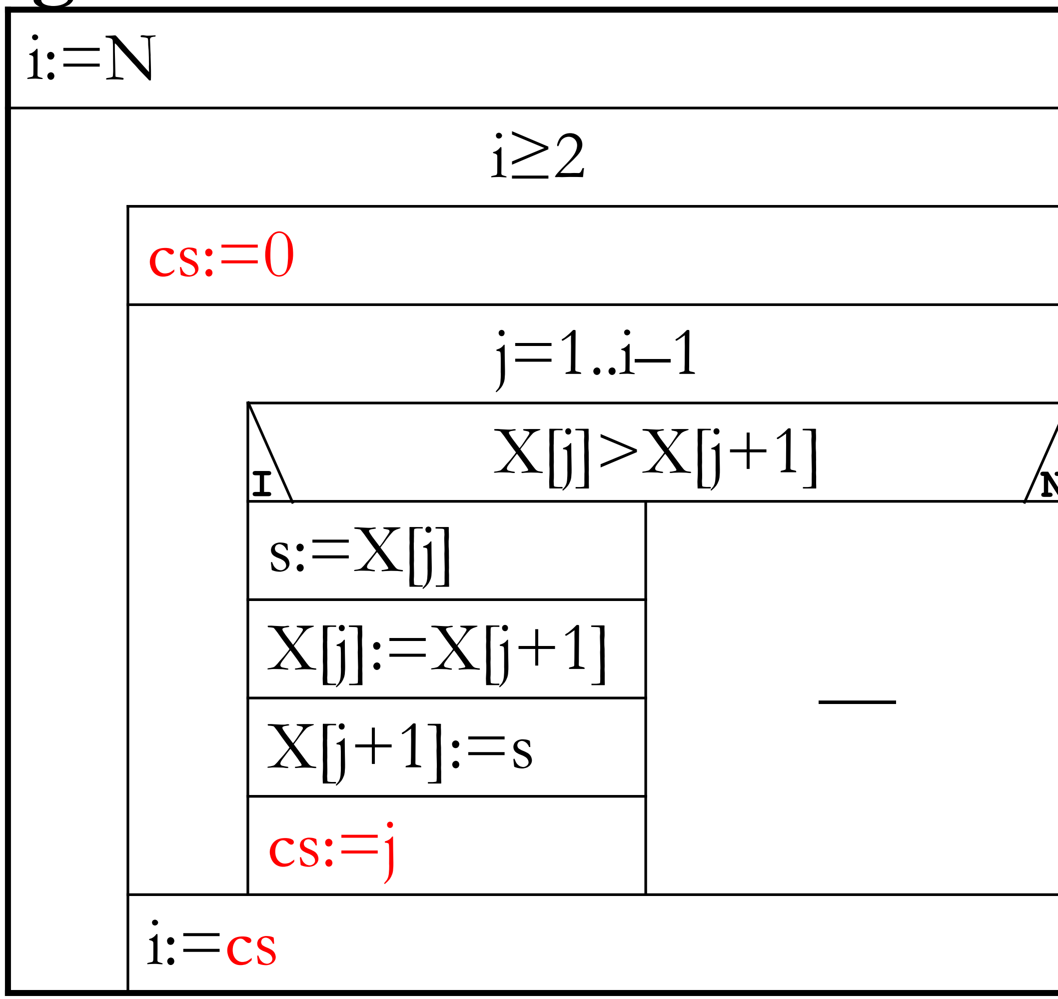
- Ha a belső ciklusban egyáltalán nincs csere, akkor be lehetne fejezni a rendezést.
- Ha a belső ciklusban a K. helyen van az utolsó csere, akkor a K+1. helytől már biztosan jó elemek vannak, a külső ciklusváltozóval többet is léphetnénk.



# Javított buborékos rendezés

## Algoritmus:



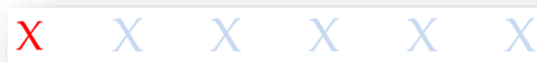




**Változó**
cs,
i,j:**Egész**
s:TH

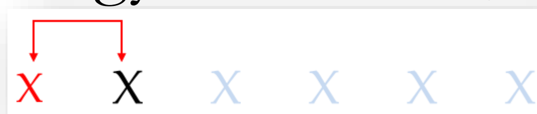
# Beillesztéses rendezés

## A lényeg:

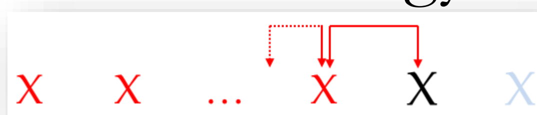
- *Egy* elem *rendezett*.




- A másodikat vagy mögé, vagy elé tesszük, így már *ketten* is *rendezettek*.




- …
- Az i-ediket a kezdő, i–1 *rendezett*ben addig hozzuk előre **cserékkel**, amíg a helyére nem kerül; így már *i darab rendezett* lesz.




- …
- Az utolsóval ugyanígy!


# Beillesztéses rendezés

## Algoritmus:

**Változó**
i,j:**Egész**
s:TH

Keresés tétel

Elem-csere

```
i=2..N
    j:=i–1
    j>0 és X[j]>X[j+1]
        s:=X[j]
        X[j]:=X[j+1]
        X[j+1]:=s
        j:=j–1
```

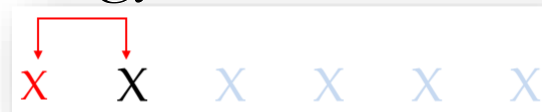
- Hasonlítások száma: N–1 .. $N \cdot \frac{N-1}{2}$
- Mozgatások száma: 0 .. $3 \cdot N \cdot \frac{N-1}{2}$

# Javított beillesztéses rendezés

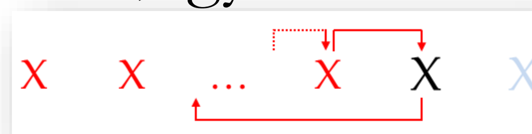
## A lényeg:

- *Egy* elem *rendezett*.

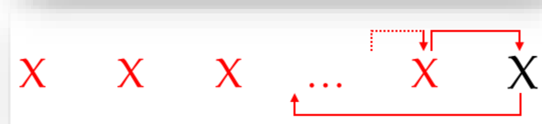

- A másodikat vagy mögé, vagy elé tesszük, így már *ketten* is *rendezettek*.




- …
- Az i-ediknél a nála nagyobbakat **tologassuk** hátra, majd illesszük be eléjük az i-ediket; így már *i darab rendezett* lesz.




- …
- Az utolsóval ugyanígy!




# Javított beillesztéses rendezés

## Algoritmus:

**Változó**
i,j:**Egész**
s:TH

i=2..N
- s:=X[i]
- j:=i–1
- j>0 és X[j]>s
  - X[j+1]:=X[j]
  - j:=j–1
- X[j+1]:=s

Elem-mozgatás, nem csere!

i=2..N
- j:=i–1
- j>0 és X[j]>X[j+1]
  - S:=X[j]
  - X[j]:=X[j+1]
  - X[j+1]:=S
  - j:=j–1

Hasonlítások száma: N–1 … $N \cdot \frac{N-1}{2}$

Mozgatások száma: 0 … $3 \cdot N \cdot \frac{N-1}{2}$

- Hasonlítások száma: N–1 .. $N \cdot \frac{N-1}{2}$
- Mozgatások száma: 2·(N–1) .. $(N+4) \cdot \frac{N-1}{2}$

# Nemhelyben rendezések

**A lényeg:**

A bemeneti és a kimeneti sorozat **eltér**.

**Specifikáció**:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{H}^N$
- Kimenet: $Y_{1..N} \in \mathbb{H}^N$

# Szétosztó rendezés

**A lényeg:**

Ha a rendezendő sorozatról speciális tudásunk van, akkor megpróbálkozhatunk más módszerekkel is.

**Specifikáció** – rendezés N lépésben:

- Bemenet: $N \in \mathbb{N}, X_{1..N} \in \mathbb{Z}^N$
- Kimenet: $Y_{1..N} \in \mathbb{Z}^N$
- Előfeltétel: $X \in$ Permutáció(1,…,N)
- Utófeltétel: RendezettE(Y) és $Y \in$ Permutáció(X)

Specialitás: a rendezett hely azonos az értékkel (vagy abból egyszerűen meghatározható).

# Szétosztó rendezés

**Algoritmus** (másolás tétel)**:**

| i=1..N |
|---|
| Y[X[i]]:=X[i] |

**Változó**
**i:Egész**

- Persze ezt írhattuk volna így is: Y[i]:=i! ☺
  Azaz a feladat akkor érdekes, ha X[i] egy **rekord**ként ábrázolható, amelynek csak **egyik mezője** (kulcsa) az 1 és N közötti egész szám:
  X,Y:Tömb[1..N:Rekord(kulcs:**1..N**,…)]

**Algoritmus** (másolás tétel)**:**

| i=1..N |
|---|
| Y[X[i].kulcs]:=X[i] |

**Változó**
**i:Egész**

# Számlálva szétosztó rendezés

## Előfeltétel:

A rendezendő értékek 1 és M közötti egész számok, továbbá ismétlődhetnek.

## Specifikáció:

- Bemenet: $N,M \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}^N$
- Kimenet: $Y_{1..N} \in \mathbb{Z}^N$
- Előfeltétel: $M \geq 1$ és $\forall i(1 \leq i \leq N)$: $1 \leq X_i \leq M$
- Utófeltétel: RendezettE(Y) és $Y \in$ Permutáció(X)

# Számlálva szétosztó rendezés

## A lényeg:

- Első lépésben számláljuk meg, hogy melyik értékből hány van a rendezendő sorozatban!
  (*megszámolás*)
- Ezután adjuk meg, hogy az első „i” értéket hova kell tenni: ez pontosan az i-nél kisebb számok száma a sorozatban +1!
  (*rekurzív kiszámítás*)
- Végül nézzük végig újra a sorozatot, s az „i” értékű elemet tegyük a helyére, majd módosítsunk: az első i értékű elemet ettől kezdve eggyel nagyobb helyre kell tenni. (*másolás*)

# Számlálva szétosztó rendezés

## Algoritmus:

Db[i]: hány darab van i-ből?

Első[i]: hol az i. elsője?

| |
|---|
| Db[1..M]:=0 |
| i=1..N |
|     Db[X[i]]:=Db[X[i]]+1 |
| Első[1]:=1 |
| i=1..M-1 |
|     Első[i+1]:=Első[i]+Db[i] |
| i=1..N |
|     Y[Első[X[i]]]:=X[i] |
|     Első[X[i]]:=Első[X[i]]+1 |

**Változó**
i:**Egész**
Db,
Első:Tömb[…]

- Mozgatások száma: N
- Additív műveletek száma: 2·M–2+2·N

# Számlálva szétosztó rendezés

## Algoritmus:

| |
|---|
| Db[1..M]:=0 |
| i=1..N |
|     Db[X[i]]:=Db[X[i]]+1 |
| Első[1]:=1 |
| i=1..M-1 |
|     Első[i+1]:=Első[i]+Db[i] |
| i=1..N |
|     Y[Első[X[i]]]:=X[i] |
|     Első[X[i]]:=Első[X[i]]+1 |

**Változó**
  i:**Egész**
  Db,
  Első:Tömb[…]

Az alaphalmaz a Z, így a többi értékadást – mint mozgatást – is beleszámíthatjuk!

- Mozgatások száma: M+N+M+2·N=2·M+3·N
- Additív műveletek száma: 2·M–2+2·N

# Számláló rendezés

## A lényeg:

- Ha nem megy a számlálva szétosztó rendezés (ismeretlen az M, vagy M»$N^2$), akkor először számláljunk *(=határozzuk meg a sorrendet)*, csak azután osszunk szét *(=tegyünk helyre…)*!
- A számláláshoz használhatjuk az egyszerű cserés rendezés elvét.
- Jelentse Db[i] az i. elemnél kisebb, vagy az i.-kel egyenlő, de tőle balra levő elemek számát!

↓

A Db[i]+1 használható az i. elemnek a rendezett sorozatbeli indexeként.

- Ehhez használhatjuk a legegyszerűbb, cserés rendezés elvét.
- Jelentse Db[i] az i. elemnél kisebb, vagy az i.-kel egyenlő, de tőle balra levő elemek számát!

# Számláló rendezés

## Algoritmus:

Az egyszerű cserés rendezés elvén működő számlálás.

```
i=1..N–1
  j=i+1..N
    X[i]>X[j]
      I: S:=X[i]
         X[i]:=X[j]
         X[j]:=S
      N: —
```

Változó
  i,j:**Egész**
  S:TH

Másolás tétel

```
Db[1..N]:=0
i=1..N–1
  j=i+1..N
    X[i]>X[j]
      I: Db[i]:=Db[i]+1
      N: Db[j]:=Db[j]+1
i=1..N
  Y[Db[i]+1]:=X[i]
```

**Változó**
  i,j:**Egész**
  Db:Tömb[.

- Hasonlítások száma: 1+2+..+N–1= $N \cdot \frac{N-1}{2}$
- Mozgatások száma: N
- Additív műveletek száma: ~hasonlítások száma

# Rendezések hatékonysága

## $N^2$ idejű rendezések:

- Egyszerű cserés rendezés
- Minimum-kiválasztásos rendezés
- Buborékos rendezés
- Javított buborékos rendezés
- Beillesztéses rendezés
- Javított beillesztéses rendezés
- Számláló rendezés

# Rendezések hatékonysága

## N (N+M) idejű rendezések:

(de speciális feltétellel)

- Szétosztó rendezés
- Számlálva szétosztó rendezés

## Kitekintés: (Algoritmusok tantárgy)

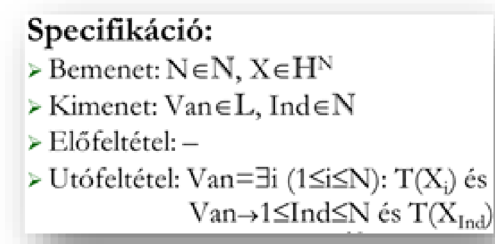
- Lesznek N·log(N) idejű rendezések.
- Nincs N·log(N)-nél jobb hasonlításokon alapuló rendezés!
- https://www.youtube.com/watch?v=ZZuD6iUe3Pc
- http://www.sorting-algorithms.com/
- e-tananyag

# Keresés rendezett sorozatban

## Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

## Specifikáció:


**Specifikáció:**
- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: –
- Utófeltétel: $Van = \exists i\,(1 \le i \le N): T(X_i)$ és $Van \to 1 \le Ind \le N$ és $T(X_{Ind})$

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in H^N$
  $Y \in H$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: RendezettE(X)
- Utófeltétel: $Van = \exists i(1 \le i \le N): X_i = Y$ és
  $Van \to 1 \le Ind \le N$ és $X_{Ind} = Y$
- Definíció (emlékeztető):
  $RendezettE(X_{1..N}) := \forall i(1 \le i < N): X_i \le X_{i+1}$

T-tulajdonság:
$T(x) := (x = Y)$

Konkretizáljuk:
legyen növekvő!

# Keresés rendezett sorozatban

## Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

## Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in H^N$
  $Y \in H$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: RendezettE(X)
- Utófeltétel: $$(Van,Ind)=\underset{\substack{i=1\\X_i=Y}}{\overset{N}{\text{Keres}}}\; i$$
- Definíció (emlékeztető):
  $$\text{RendezettE}(X_{1..N}):=\forall i(1\leq i<N): X_i\leq X_{i+1}$$

**Specifikáció:**
- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: –
- Utófeltétel: $Van=\exists i\ (1\leq i\leq N): T(X_i)$ és
  $Van \rightarrow 1\leq Ind\leq N$ és $T(X_{Ind})$

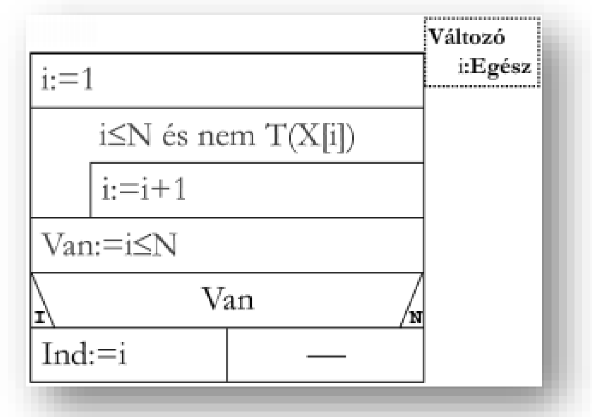
T-tulajdonság:
$T(x):=(x=Y)$

Konkretizáljuk:
legyen növekvő!

# Keresés rendezett sorozatban

## Ötlet:

Ha már a keresett elem értékénél nagyobbnál tartunk, akkor biztos nem lesz a sorozatban, megállhatunk.




Változó
i:Egész

| |
|---|
| i:=1 |
| i≤N és X[i]<Y |
|     i:=i+1 |
| Van:= i≤N és X[i]=Y |
| … |

## Észrevétel:

Van megoldás ↔ azért álltunk meg keresés közben, mert megtaláltuk a keresett értéket.

# Keresés rendezett sorozatban

## Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in H^N$
  $Y \in H$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: $N>0$ és $RendezettE(X)$
- Utófeltétel: $Van = \exists i(1 \leq i \leq N)\text{: } X_i = Y$ és
  $Van \rightarrow 1 \leq Ind \leq N$ és $X_{Ind} = Y$

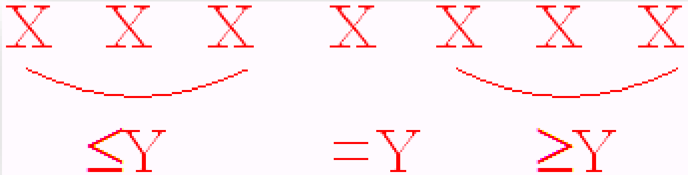
**Programparaméterek:**

**Konstans**
  MaxN:**Egész**(???)
**Típus**
  THk=**Tömb**[1..MaxN:TH]
**Változó**
  N:**Egész**, X:THk
  Y:TH
  Van:**Logikai**, Ind:**Egész**

## Ötlet és – tömb esetén – lehetőség:

Először a **középső** elemmel hasonlítsunk! Ha nem a keresett, akkor vagy **előtte**, vagy **mögötte** kell tovább keresni!

# Keresés rendezett sorozatban




## Algoritmus:

Itt akkor van megoldás, ha megtaláltuk a keresett érték valamelyikét.

**Változó**
e,k,u:**Egész**

```
e:=1
u:=N
    k:=(e+u) Div 2
    \ X[k]>Y        \ X[k]<Y
    u:=k–1          e:=k+1
e≤u és X[k]≠Y
Van:=X[k]=Y
…
```

**Specifikáció:**

- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
  $Y \in H$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: $N>0$ és RendezettE(X)
- Utófeltétel: $Van = \exists i(1 \le i \le N)$: $X_i = Y$ és
  $Van \rightarrow 1 \le Ind \le N$ és $X_{Ind} = Y$

# Keresés rendezett sorozatban

## További kérdések – tételvariánsok:

- Hány lépés alatt találjuk meg a keresett elemet? (→Logaritmikus v. bináris keresés.)
- Ha több egyforma elem is van a sorozatban, akkor ez a módszer melyiket találja meg?
- Hogyan lehetne az összes Y-értékű elemet megtalálni?

# Rendezettek uniója

Összefuttatás.

## Feladat:

Adott két rendezett halmaz, adjuk meg az uniójukat!

## Specifikáció:

- Bemenet: $N,M \in \mathbb{N}$, $X_{1..N} \in H^N$, $Y_{1..M} \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z_{1..N+M} \in H^{N+M}$ Db-ig kitöltve
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)

# Rendezettek uniója

- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és

  $\forall i(1 \leq i \leq Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és

  HalmazE(Z) és RendezettE(Z)
- Utófeltétel$_2$: (Db,Z)=Unió(N,X,M,Y) és RendezettE(Z)

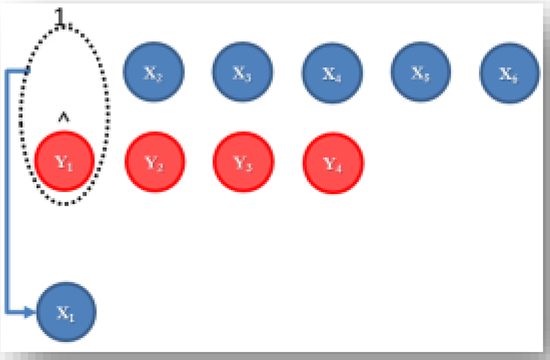
**Ötlet:**

Az eredmény első eleme vagy az X, vagy az Y első eleme lehet. A kettő közül a rendezettség szerintit tegyük az eredménybe, majd a maradékra ugyanezt az elvet alkalmazhatjuk.

# Rendezettek uniója

## Algoritmus elé:

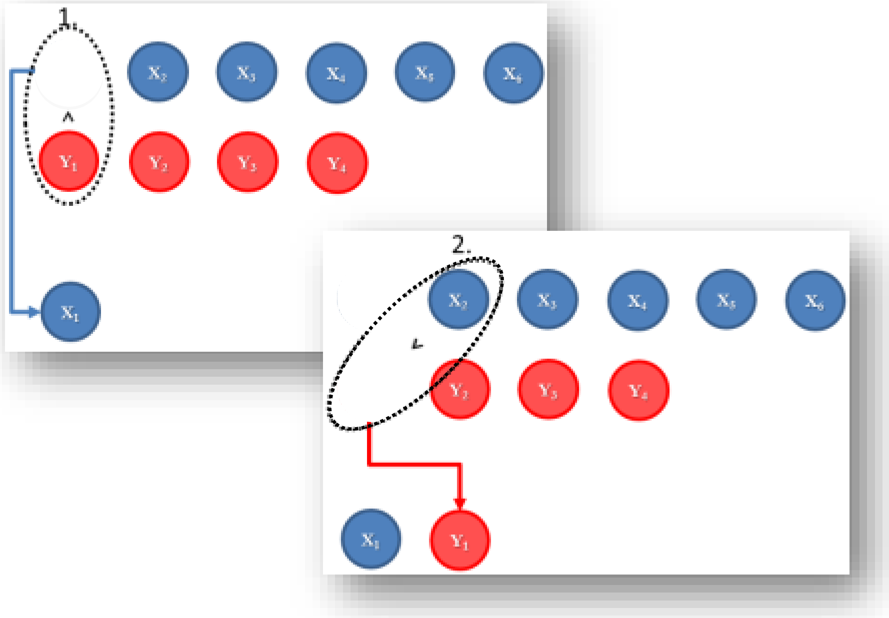
- Amíg van mit hasonlítani:



# Rendezettek uniója

## Algoritmus elé:

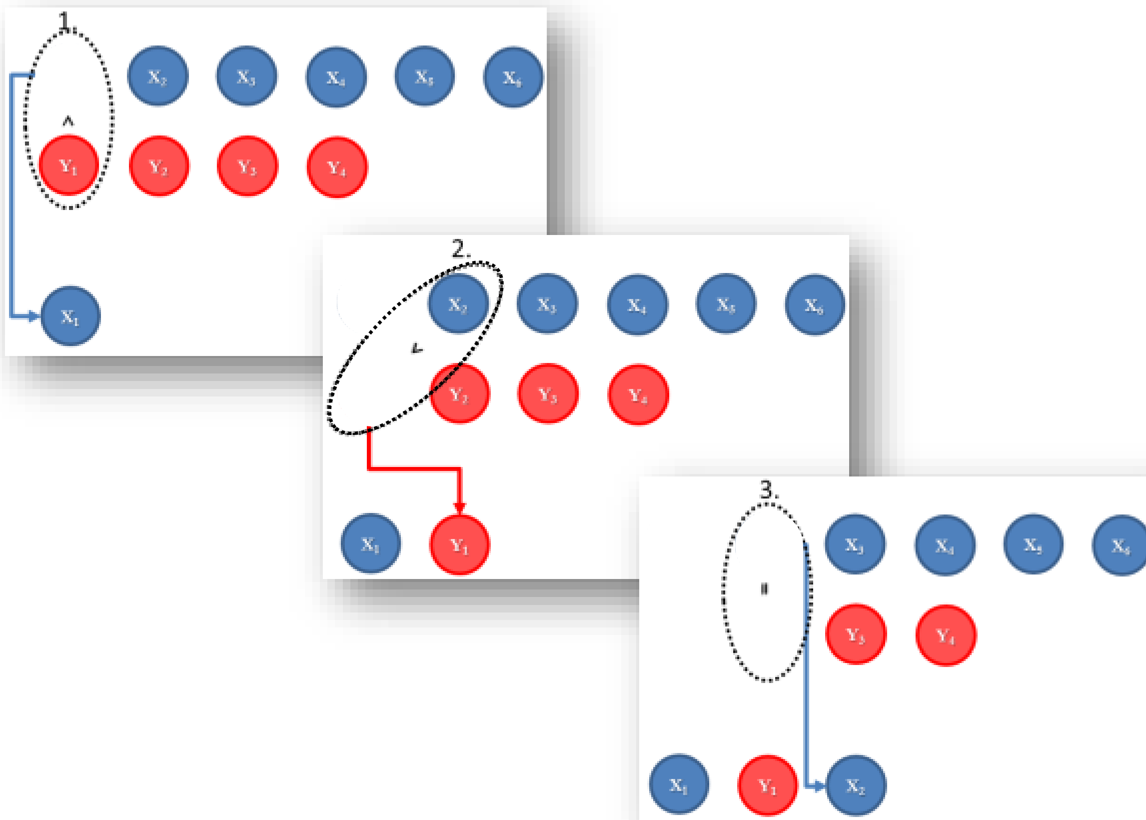
- Amíg van mit hasonlítani:



# Rendezettek uniója

## Algoritmus elé:

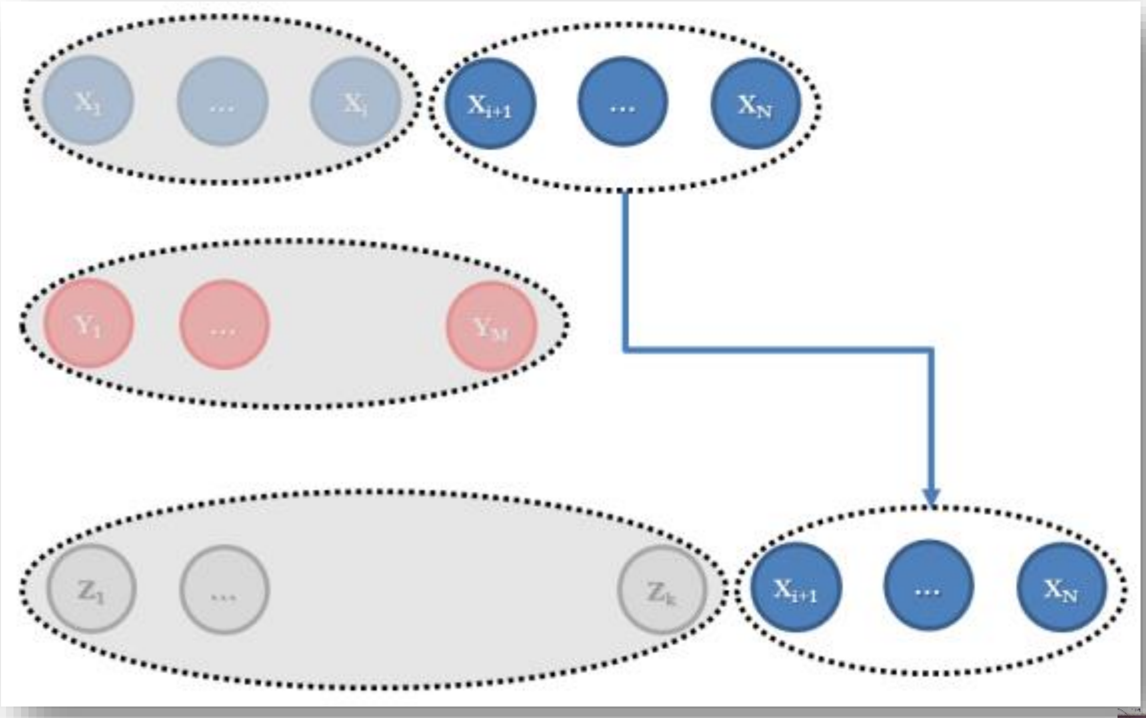
- Amíg van mit hasonlítani:



# Rendezettek uniója

## Algoritmus elé:

- Ha már nincs mit hasonlítani:



# Rendezettek uniója

## Algoritmus$_1$:

**Specifikáció:**
- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^{M} 1$ és $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

```
Z:=X
Db:=N
j=1..M
    i:=1
    i≤N és X[i]≠Y[j]
        i:=i+1
    i>N
      Db:=Db+1      |  —
      Z[Db]:=Y[j]   |
```

```
i:=1
j:=1
Db:=0
i≤N és j≤M
    Db:=Db+1
    X[i]<Y[j]      | X[i]=Y[j]      | X[i]>Y[j]
    Z[Db]:=X[i]    | Z[Db]:=X[i]    | Z[Db]:=Y[j]
    i:=i+1         | i:=i+1         | j:=j+1
                   | j:=j+1         |
…
```

Van miket hasonlítani

**Változó**
i,j:Eg

# Rendezettek uniója

## Algoritmus$_1$:

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és $\forall i (1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

Változó
i,j:Eg

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| i≤N és j≤M | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |
| … | | |

Van miket hasonlítani

| | |
|---|---|
| Z:=X | |
| Db:=N | |
| j=1..M | |
| i:=1 | |
| i≤N és X[i]≠Y[j] | |
| i:=i+1 | |
| i>N | |
| Db:=Db+1 | — |
| Z[Db]:=Y[j] | |

# Rendezettek uniója

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és
  $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és
  HalmazE(Z) és RendezettE(Z)

…

i≤N

Db:=Db+1

Z[Db]:=X[i]

i:=i+1

j≤M

Db:=Db+1

Z[Db]:=Y[j]

j:=j+1

Nincs Y-beli.

Nincs X-beli.

Z:=X

Db:=N

j=1..M

i:=1

i≤N és X[i]≠Y[j]

i:=i+1

i>N

Db:=Db+1 | —

Z[Db]:=Y[j]

# Rendezettek uniója

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és $\forall i (1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

Nincs Y-beli.

Nincs X-beli.

…

$i \le N$

- Db:=Db+1
- Z[Db]:=X[i]
- i:=i+1

$j \le M$

- Db:=Db+1
- Z[Db]:=Y[j]
- j:=j+1

| i≤N és j≤M | | |
|---|---|---|
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | X[i]>Y[j] |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

Vegyük észre: ha az X és Y utolsó elemei egyenlők, akkor ez a két ciklus nem kell!

# Rendezettek uniója

## Algoritmus$_2$:

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| X[N+1]:=+∞ | | |
| Y[M+1]:=+∞ | | |
| i≤N+1 és j≤M+1 | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | X[i]>Y[j] |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

… és utoljára?
Z[Db]:=+ ∞

Változ
i,j:E

# Rendezettek uniója

## Algoritmus$_2$:

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

Változ
i,j:E

| |
|---|
| i:=1 |
| j:=1 |
| Db:=0 |
| X[N+1]:=+∞ |
| Y[M+1]:=+∞ |
| i≤N+1 és j≤M+1 |

| X[i]<Y[j] | X[i]=Y[j] | |
|---|---|---|
| Db:=Db+1 | | |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

… és utoljára?
Z[Db]:=+ ∞

# Rendezettek uniója

## Algoritmus$_2$ javítása:

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N}$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és
  $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| X[N+1]:=+∞ | | |
| Y[M+1]:=+∞ | | |
| i≤N+1 és j≤M+1 | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | X[i]>Y[j] |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| X[N+1]:=+∞ | | |
| Y[M+1]:=+∞ | | |
| i<N+1 vagy j<M+1 | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

**Változ**
i,j:E

# Rendezettek uniója

## Algoritmus$_2$ javítása:

**Specifikáció:**

- Bemenet: $N,M \in \mathbb{N},\ X \in H^N,\ Y \in H^M$
- Kimenet: $Db \in \mathbb{N},\ Z \in H^{Db}$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)
- Utófeltétel$_1$: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{M} 1$ és $\forall i(1 \le i \le Db)$: $Z_i \in X$ vagy $Z_i \in Y$ és HalmazE(Z) és RendezettE(Z)

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| X[N+1]:=+∞ | | |
| Y[M+1]:=+∞ | | |
| i≤N+1 és j≤M+1 | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | X[i]>Y[j] |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

| | | |
|---|---|---|
| i:=1 | | |
| j:=1 | | |
| Db:=0 | | |
| X[N+1]:=+∞ | | |
| Y[M+1]:=+∞ | | |
| i≤N vagy j≤M | | |
| Db:=Db+1 | | |
| X[i]<Y[j] | X[i]=Y[j] | |
| Z[Db]:=X[i] | Z[Db]:=X[i] | Z[Db]:=Y[j] |
| i:=i+1 | i:=i+1 | j:=j+1 |
| | j:=j+1 | |

Változ
i,j:E

# Rendezettek uniója

```
i:=1
j:=1
Db:=0
i≤N és j≤M
  Db:=Db+1
  X[i]<Y[j]    | X[i]=Y[j]    |
  Z[Db]:=X[i]  | Z[Db]:=X[i]  | Z[Db]:=Y[j]
  i:=i+1       | i:=i+1       | j:=j+1
               | j:=j+1       |
i≤N
  Db:=Db+1
  Z[Db]:=X[i]
  i:=i+1
j≤M
  Db:=Db+1
  Z[Db]:=Y[j]
  j:=j+1
```

```
i:=1
j:=1
Db:=0
X[N+1]:=+∞
Y[M+1]:=+∞
i≤N vagy j≤M
  Db:=Db+1
  X[i]<Y[j]    | X[i]=Y[j]    |
  Z[Db]:=X[i]  | Z[Db]:=X[i]  | Z[Db]:=Y[j]
  i:=i+1       | i:=i+1       | j:=j+1
               | j:=j+1       |
```

## Kérdések:

- Jobb lett ez a módszer az előzőnél az idő szempontból?
  ⇐ Hány lépés alatt kapjuk meg a megoldást?
- Meg lehetne ugyanezt tenni a metszettel is?

## Tapasztalat:

- Jobb lett ez a módszer bonyolultság szempontjából. (⇐ Ciklus-/elágazás-szám.)
- Ez a módszer a kimenet szerint halad egyesével és nem a bemenet szerint (mint a korábbiak).

# Rendezettek összefésülése

**Feladat:**

Adott két rendezett sorozat, adjuk meg az összefésülésüket!

**Specifikáció:**

- Bemenet: $N,M \in N$, $X_{1..N} \in H^{N}$, $Y_{1..M} \in H^{M}$
- Kimenet: $Z_{1..N+M} \in H^{N+M}$
- Előfeltétel: ~~HalmazE(X) és HalmazE(Y) és~~
  RendezettE(X) és RendezettE(Y)

# Rendezettek összefésülése

- Utófeltétel: $Z \in$ Permutáció($X \oplus Y$) és
  RendezettE($Z$)

**Ötlet:**

A megoldás olyan, mint az összefuttatás, csak az egyforma elemeket is berakjuk az eredménybe, tehát egy-egy érték multiplicitása lehet 1-nél nagyobb is (már kezdetben is!).

# Rendezettek összefésülése

## Algoritmus:

**Specifikáció:**

- Bemenet: $N,M \in N$, $X \in H^N$, $Y \in H^M$
- Kimenet: $Z \in H^{N+M}$
- Előfeltétel: RendezettE(X) és RendezettE(Y)
- Utófeltétel: $Z \in$ Permutáció($X \oplus Y$) és RendezettE(Z)

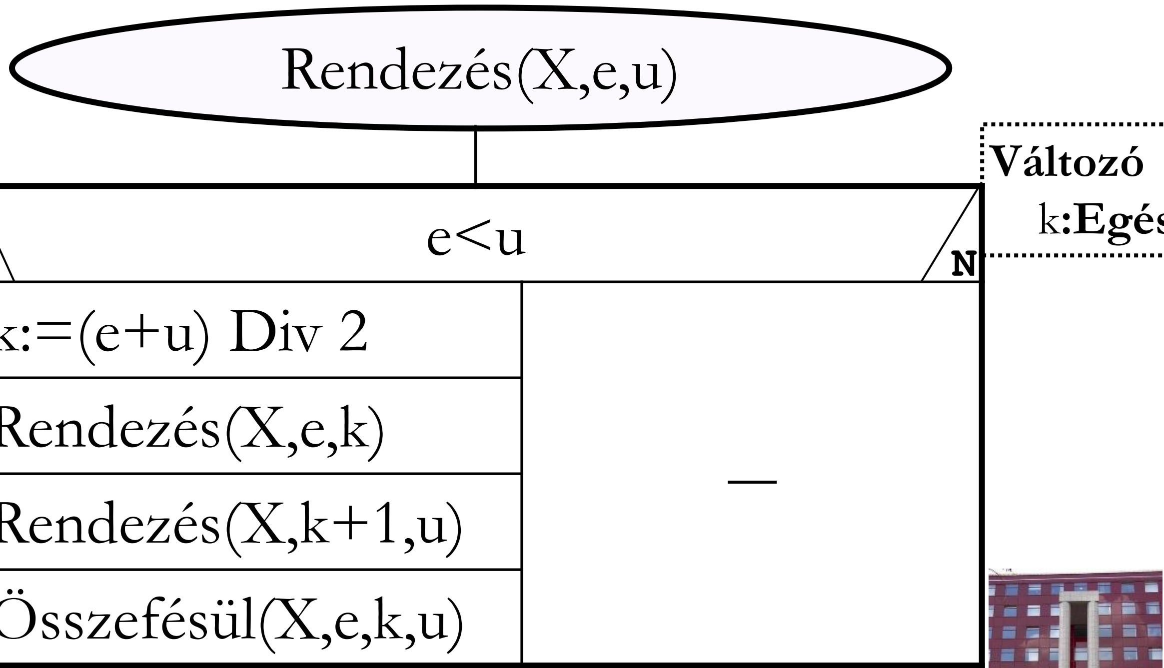
**Változó**
i,j:**Egész**

```
i:=1
j:=1
Db:=0
X[N+1]:=+∞
Y[M+1]:=+∞
i≤N vagy j≤M
    Db:=Db+1
    X[i] ≤ Y[j]
    I: Z[Db]:=X[i]      N: Z[Db]:=Y[j]
       i:=i+1              j:=j+1
```

# Összefésüléses rendezés

## Ötlet:

Az összefésülés elvére alapozhatjuk az összefésüléses rendezést: amennyiben egy sorozat nem egyelemű, akkor középen vágjuk ketté, mindkét felét rendezzük (rekurzívan), majd a két rendezett sorozatot fésüljük össze!




# Oszd meg és uralkodj!

Az előző algoritmus (illetve a logaritmikus keresés) alapján megfogalmazhatunk egy általános tervezési elvet:
Több részfeladatra bontás, amelyek hasonlóan oldhatók meg, lépései:

- a triviális eset (amikor nincs rekurzív hívás)
- felosztás (megadjuk a részfeladatokat, amikre a feladat lebontható)
- uralkodás (rekurzívan megoldjuk az egyes részfeladatokat)
- összevonás (az egyes részfeladatok megoldásából előállítjuk az eredeti feladat megoldását)

# Oszd meg és uralkodj!

Ezek alapján a következőképpen fogunk gondolkodni:

- Mi a leállás (triviális eset) feltétele? Hogyan oldható meg ilyenkor a feladat?
- Mi az általános feladat alakja? Mik a paraméterei? Ebből kapjuk meg a rekurzív eljárásunk specifikációját.
- Milyen paraméter értékekre kapjuk a konkrét feladatot? Ezekre fogjuk meghívni kezdetben az eljárást!
- Hogyan vezethető vissza a feladat hasonló, de egyszerűbb részfeladatokra? Hány részfeladatra vezethető vissza?
- Melyek ilyenkor az általános feladat részfeladatainak a paraméterei? Ezekkel kell majd meghívni a rekurzív eljárást!
- Hogyan építhető fel a részfeladatok megoldásaiból az általános feladat megoldása?

# Oszd meg és uralkodj!

A korábban megismert helyben szétválogatás algoritmusra építhetjük ezen az elven a gyorsrendezés algoritmusát:

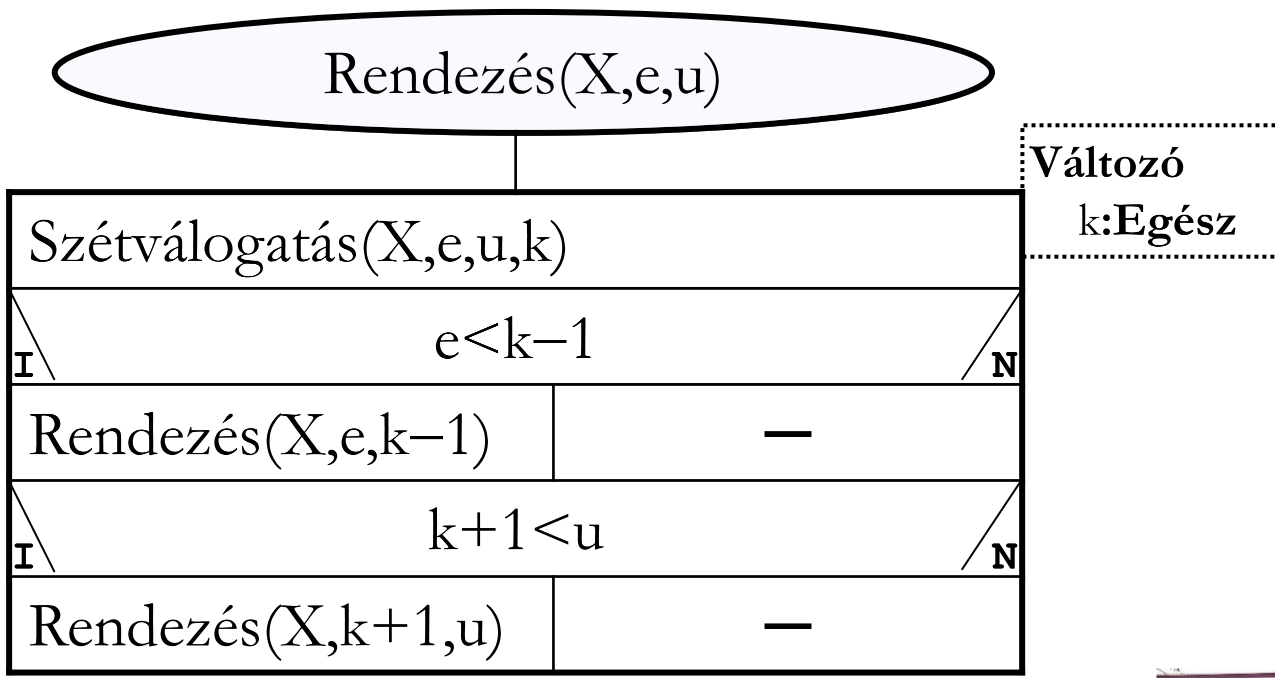
**Gyorsrendezés (quicksort):**

- felbontás: $\boxed{X_1,\ldots, X_{k-1}}$ $X_k$ $\boxed{X_{k+1},\ldots, X_n}$ szétválogatás

  ahol $\forall i,j$ $(1\leq i<k;\ k<j\leq n)$: $X_i\leq X_k$ és $X_k\leq X_j$
- uralkodás: mindkét részt ugyanazzal a módszerrel felbontjuk két részre, rekurzívan
- összevonás: automatikusan történik a helyben szétválogatás miatt
- triviális eset: $n\leq 1$

# Oszd meg és uralkodj!

## Gyorsrendezés (quicksort):




# Tartalom