

An aerial photograph of Budapest, Hungary, showing the Danube River and the city skyline. The image is in black and white, with a semi-transparent white rectangle overlaid in the center containing the text "Programozás 12. előadás".

Programozás 12. előadás

Tartalom



➤ Rendezési feladat

- Specifikáció
- Egyszerű cserés rendezés
- Minimum-kiválasztásos rendezés
- Buborékos rendezés
- Javított buborékos rendezés

- Beillesztéses rendezés
- Javított beillesztéses rendezés
- Szétoztó rendezés
- Számlálva szétoztó rendezés
- Számláló rendezés

➤ Rendezések hatékonysága – idő

➤ Algoritmusok rendezett sorozatokban

- Keresés rendezett sorozatban
- Rendezettek uniója, összefésülése
- Összefésüléssel rendezés

➤ Oszd meg és uralkodj!



Rendezési feladat

Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X_{1..N} \in H^N$
 $\leq: H \times H \rightarrow L$
- Kimenet: $Y_{1..N} \in H^N$
- Előfeltétel: **Rendezés**(\leq) és **Rendezett** $E_{\leq}(H)$
- Utófeltétel: **Rendezett** $E_{\leq}(Y)$ és $Y \in$ **Permutáció**(X)
- Jelölések:
 - **Rendezett** $E_{\leq}(X/H)$: X/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: $N \in \mathbb{N}$, $X_{1..N} \in H^N$, $\leq: H \times H \rightarrow L$
- Kimenet: $X'_{1..N} \in H^N$
- Előfeltétel: **Rendezés**(\leq) és **RendezettE** $_{\leq}$ (H)
- Utófeltétel: **RendezettE** $_{\leq}$ (X') és $X' \in$ **Permutáció**(X)
- Jelölések:
 - X': az X **kimeneti** (megálláskori) értéke
 - **RendezettE** $_{\leq}$ (X/H): X/H **rendezett-e** a \leq -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 „megapozztrofáljuk”.

Pl.: $Z' := a$ Z **kimeneti** (megálláskori) értéke.

➤ $A \leq$ reláció **rendezés**, ha

1. *reflexív*: $\forall h \in H: h \leq h$
2. *antiszimmetrikus*: $\forall h, i \in H: h \leq i \text{ és } i \leq h \rightarrow h = i$
3. *transzitív*: $\forall h, i, j \in H: h \leq i \text{ és } i \leq j \rightarrow h \leq j$



Rendezések

(fontos új fogalmak, jelölések)

- H (teljesen) **rendezett halmaz**:

$$\text{Rendezett } E: H \rightarrow L$$

$$\text{Rendezett } E(H) := \forall h, i \in H: h \leq i \text{ vagy } i \leq h$$

- **Rendezett sorozat**:

$$\text{Rendezett } E: H^* \rightarrow L$$

$$\text{Rendezett } E(Z) := \forall i (1 \leq i \leq N-1): Z_i \leq Z_{i+1}$$

- **Permutációhalmaz**:

$$\text{Permutáció}: H^* \rightarrow (H^*)^*$$

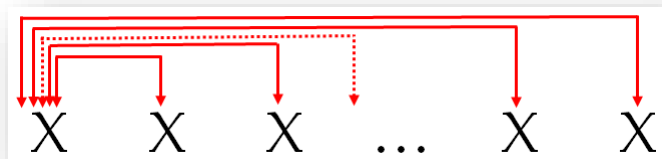
Permutáció(Z) := a $Z \in H^N$ sorozat elemeinek *összes permutációját* (sorozatot!) tartalmazó *halmaz*, amelynek tehát egyik eleme a kívánt rendezettségű sorozat...



Egyszerű cserés rendezés

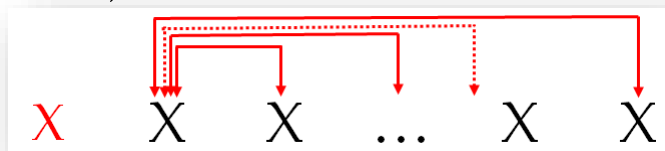
A lényeg:

- 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!



- ...

- Végül az utolsó két elemre!

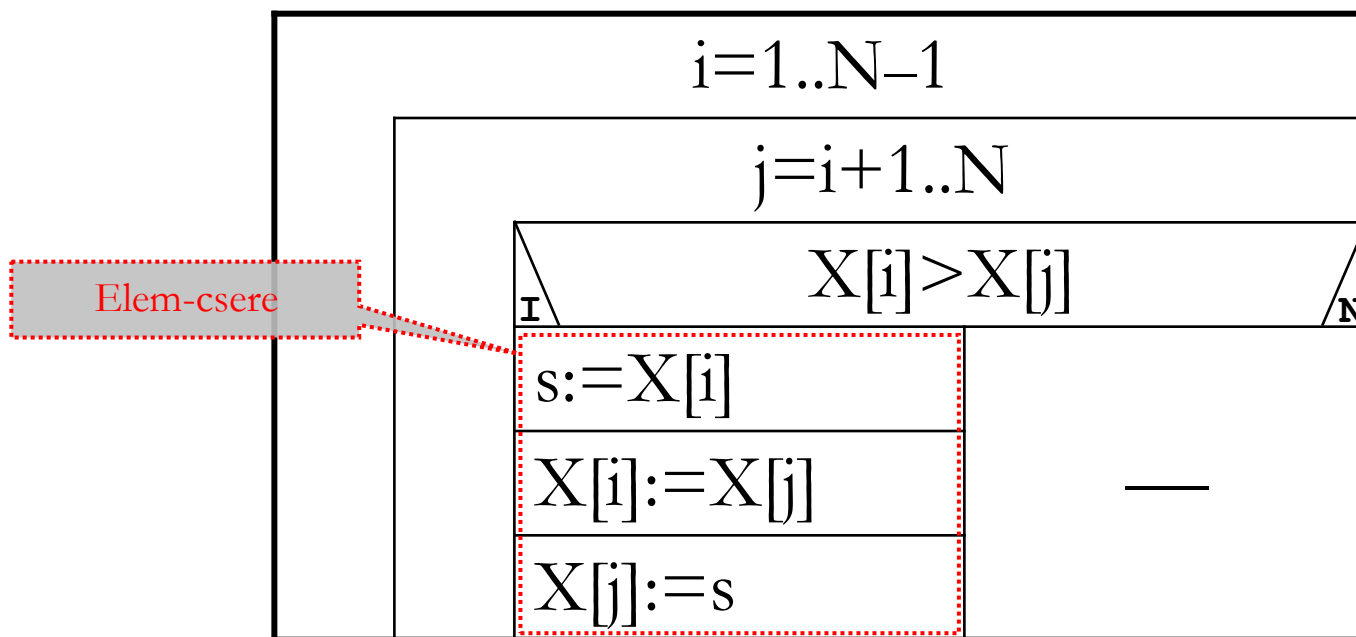


A pirossal jelöltek már a helyükön vannak



Egyszerű cserés rendezés

Algoritmus:



Változó
i,j: Egész
s: TH

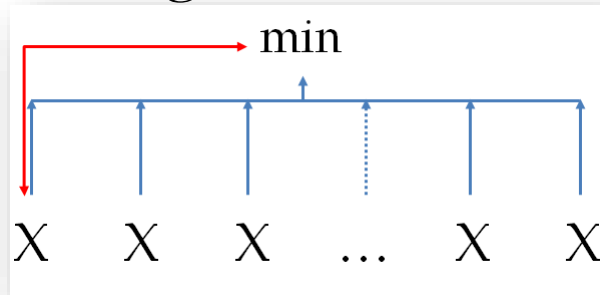
- Hasonlítások száma: $1 + 2 + \dots + N - 1 = N \cdot \frac{N - 1}{2}$
- Mozgatások száma: $0 \dots 3 \cdot N \cdot \frac{N - 1}{2}$



Minimum-kiválasztásos rendezés

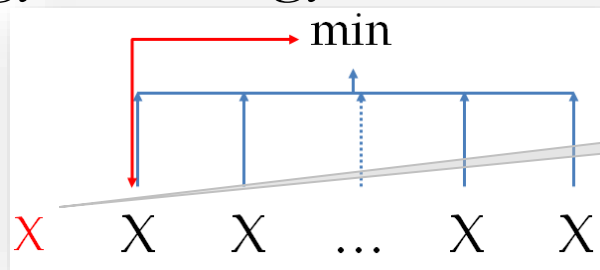
A lényeg:

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



A minimum az „alsó” végére kerül.

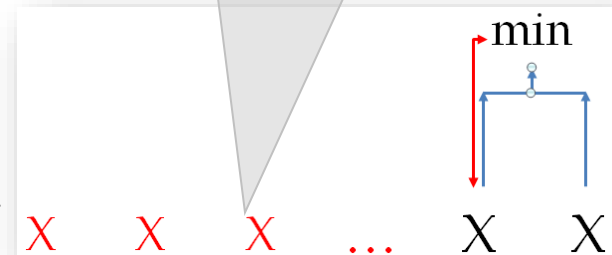
- Ezután ugyanezt tegyük a $2..N$ elemre!



A pirossal jelöltek már a helyükön vannak

- ...

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

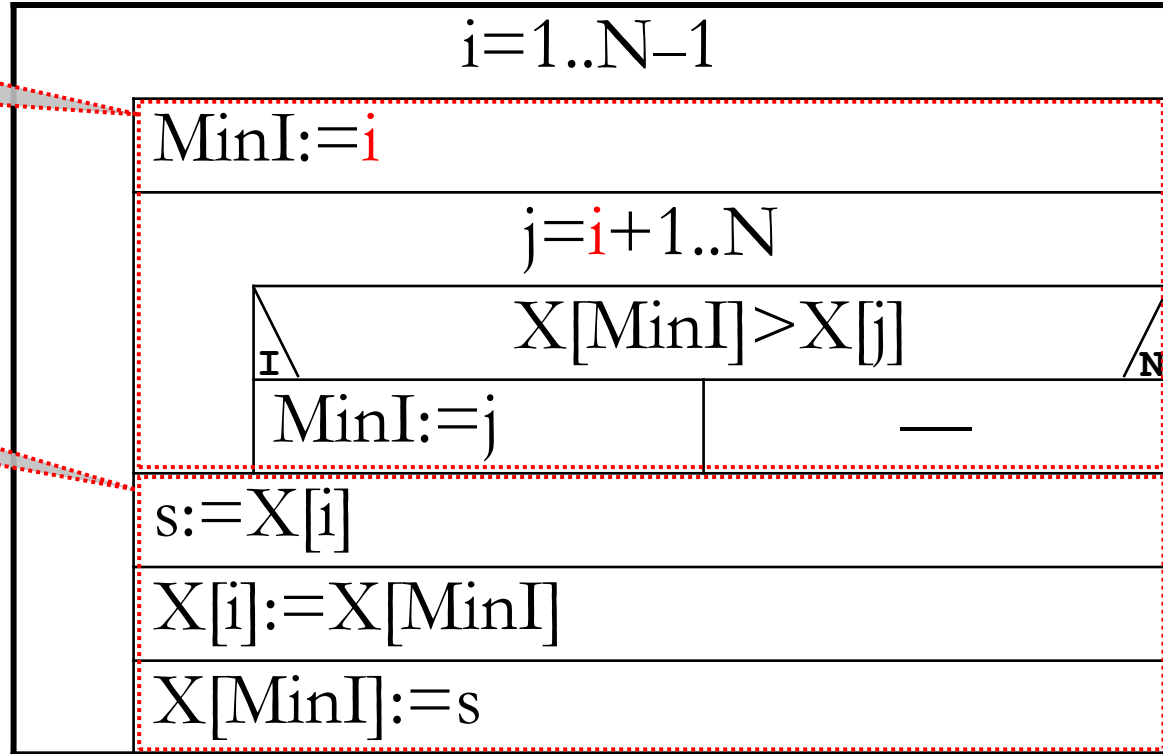


Minimum-kiválasztásos rendezés

Algoritmus:

Változó
MinI,
i,j: Egész
s: TH

Minimum-
kiválasztás az i.-től



Elem-csere

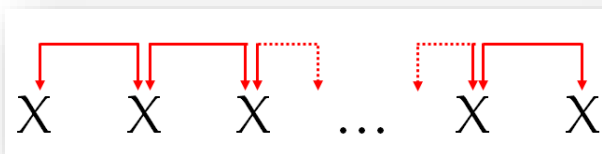
- Hasonlítások száma: $1+2+\dots+N-1=N\cdot\frac{N-1}{2}$
- Mozgatások száma: $3\cdot(N-1)$



Buborékos rendezés

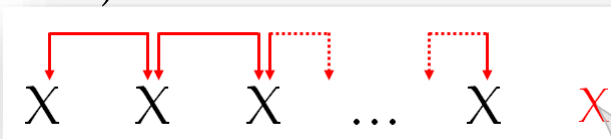
A lényeg:

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



A maximum a „felső” végére kerül.

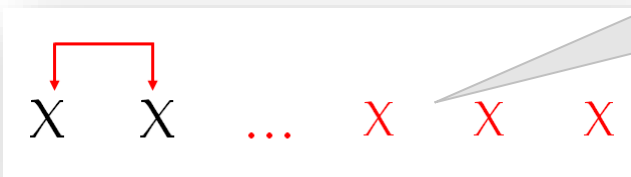
- Ezután ugyanezt csináljuk az utolsó elem nélkül!



A többiek is tartanak a helyük felé.

- ...

- Végül az első két elemre!

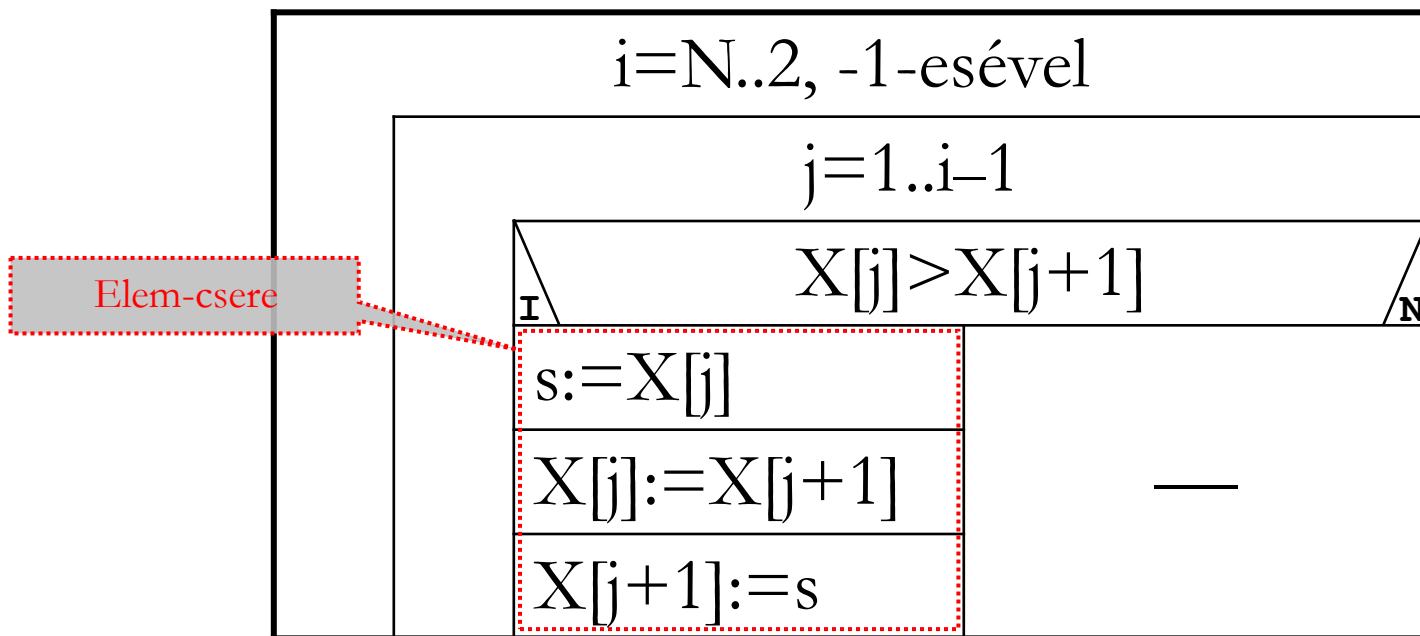


A pirossal jelöltek már a helyükön vannak



Buborékos rendezés

Algoritmus:



Változó

i, j : Egész

s : TH

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

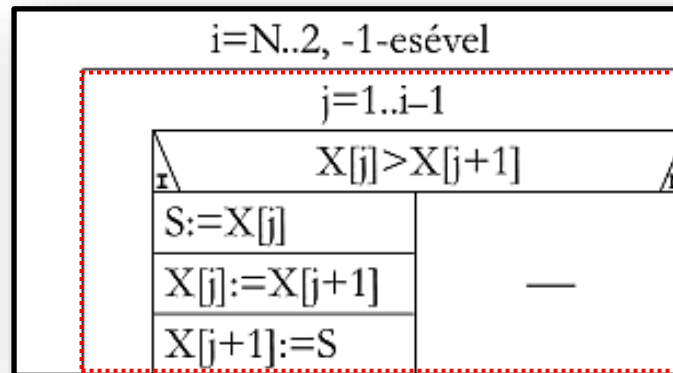


Javított buborékos rendezés



Megfigyelések:

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

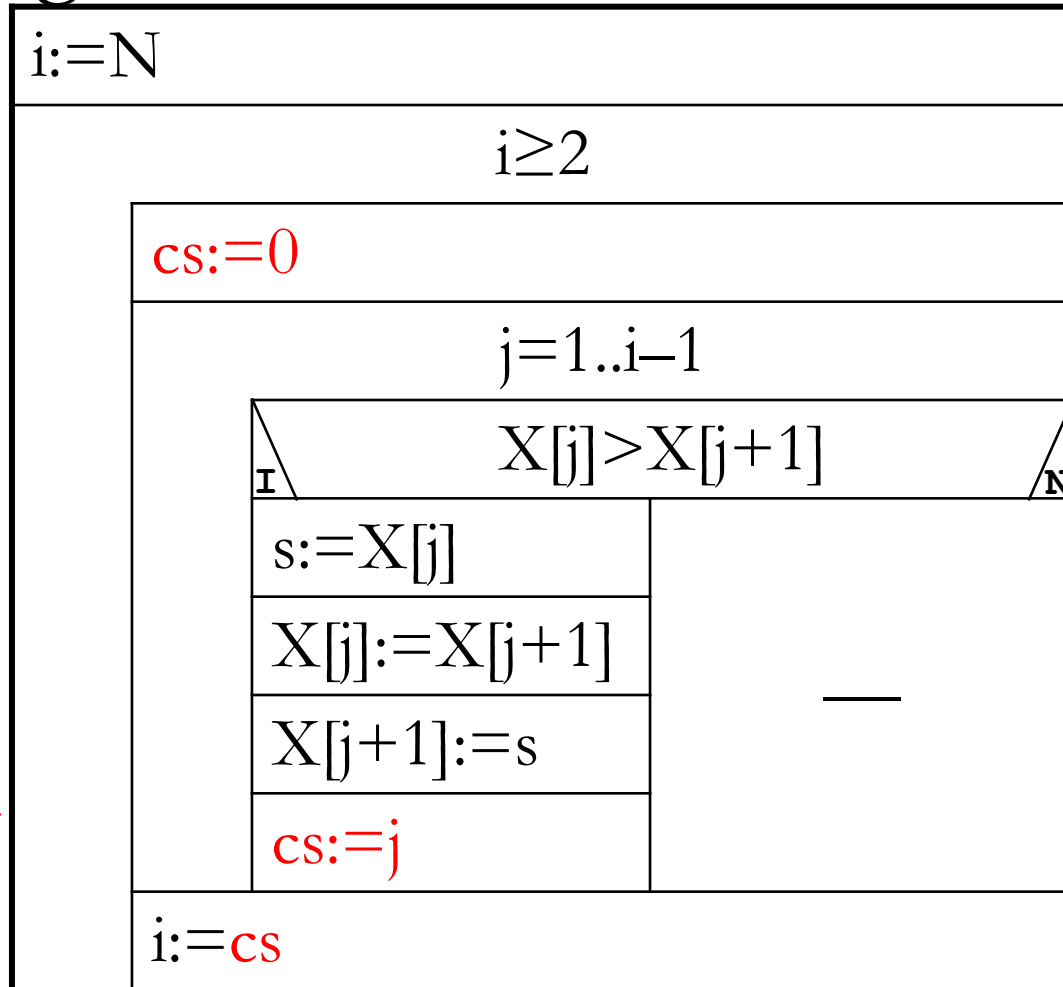


Javított buborékos rendezés

Algoritmus:

Változó

cs,
i,j:Egész
s:TH



$i = N..2, -1$ -esével

$j = 1..i-1$

$X[j] > X[j+1]$

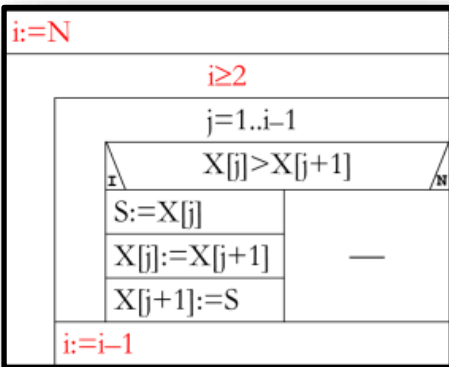
$S := X[j]$

$X[j] := X[j+1]$

$X[j+1] := S$

—

Átírás
'amíg'-os
ciklussá



Az utolsó
cseréhez
feljegyzése

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!

x x x x x x

x x x x x x

x x ... x x x

x x x ... x x



Beillesztéses rendezés

Algoritmus:

Változó

i,j:Egész

s:TH

Keresés tétel

j:=i-1

Elem-csere

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



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.

x x x x x x

x x x x x x

x x ... x x

x x x ... x x



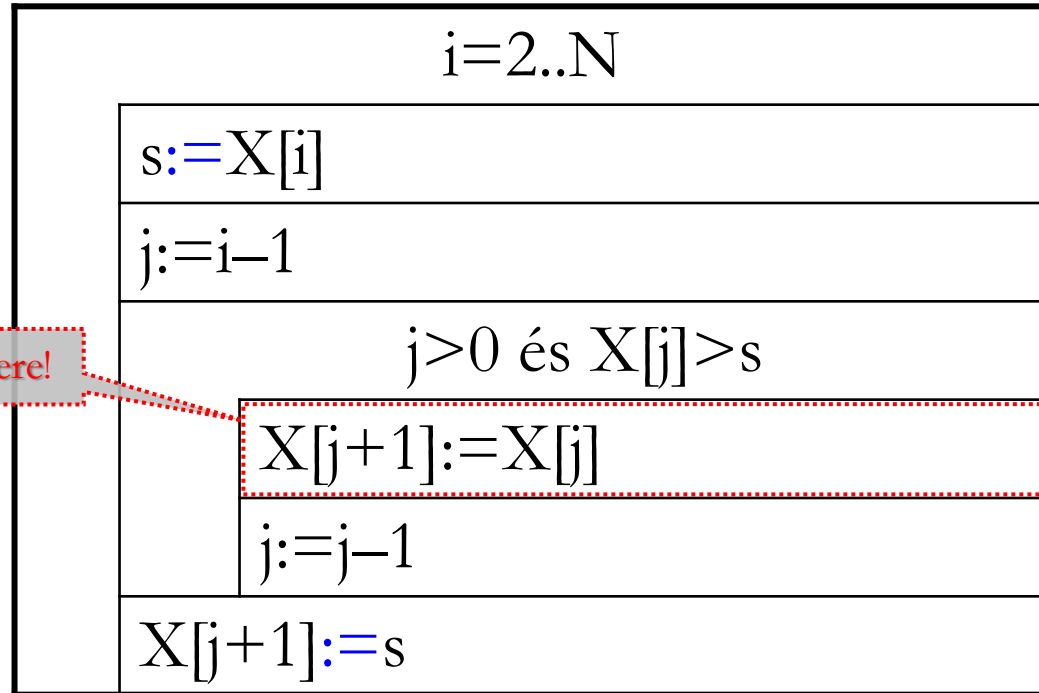
Javított beillesztéses rendezés

Algoritmus:

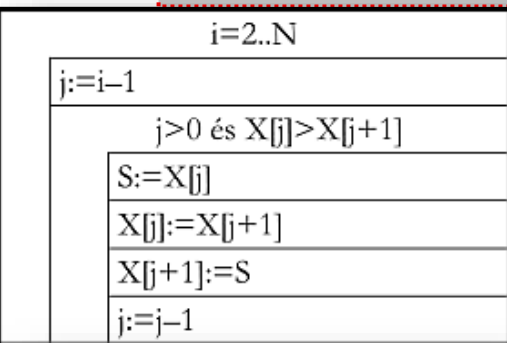
Változó

i,j:Egész

s:TH



Elem-mozgatás, nem csere!



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

➤ Hasonlítások száma: $N-1 \dots N \cdot \frac{N-1}{2}$

➤ Mozgatások száma: $2 \cdot (N-1) \dots (N+4) \cdot \frac{N-1}{2}$



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: $X \in \text{Permutáció}(1, \dots, N)$
- Utófeltétel: $\text{RendezettE}(Y)$ és $Y \in \text{Permutáció}(X)$

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 rekordként ábrázolható, amelynek csak egyik mezője (kulcsa) az 1 és N közötti egész szám:

$X, Y: \text{Tömb}[1..N: \text{Rekord}(\text{kulcs}: 1..N, \dots)]$

Algoritmus (másolás tétel):

$i=1..N$
$Y[X[i].\text{kulcs}]:=X[i]$

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: $\text{RendezettE}(Y)$ és $Y \in \text{Permutáció}(X)$



Számlálva szétosztó rendezés



A lényeg:

- Első lépésben számláljuk meg, hogy melyik értékből **hány** van a rendezendő sorozatban!
(*megszámlolás*)
- 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: $2 \cdot M - 2 + 2 \cdot N$



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 \mathbb{Z} , így a többi értékadást – mint mozgatást – is beleszámíthatjuk!

➤ Mozgatások száma: $M+N+M+2 \cdot N = 2 \cdot M+3 \cdot N$

➤ Additív műveletek száma: $2 \cdot M - 2 + 2 \cdot N$



Számláló rendezés

A lényeg:

- Ha nem megy a számlálva szétoztó rendezés (ismeretlen az M , vagy $M \gg N^2$), 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**.

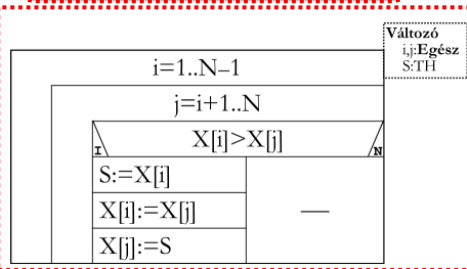


Számláló rendezés

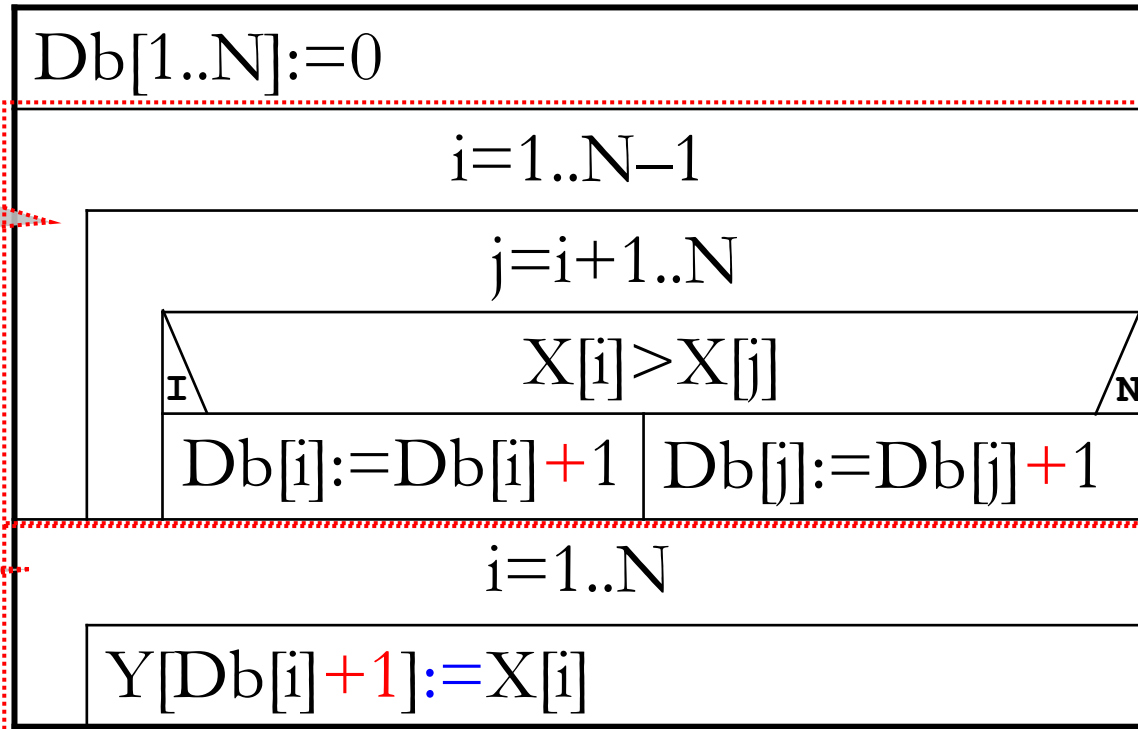
- Ehhez használhatjuk a legegyszerűbb, **cserés rendezés elvét**.
- Jelentse $Db[i]$ az i . elemnél **kisebb**, vagy az i -kel **egyenlő**, de **tőle balra levő elemek számát**!

Algoritmus:

Az egyszerű cserés rendezés elvén működő számlálás.



Másolás tétel



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 $N \cdot \log(N)$ idejű rendezések.
- Nincs $N \cdot \log(N)$ -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:

➤ Bemenet: $N \in \mathbb{N}, X_{1..N} \in H^N$

$Y \in H$

➤ Kimenet: $Van \in L, Ind \in \mathbb{N}$

➤ Előfeltétel: **RendezettE(X)**

➤ Utófeltétel: $Van = \exists i (1 \leq i \leq N): X_i = Y$ és
 $Van \rightarrow 1 \leq Ind \leq N$ és $X_{Ind} = Y$

➤ Definíció (emlékeztető):

$RendezettE(X_{1..N}) := \forall i (1 \leq i < N): X_i \leq X_{i+1}$

Specifikáció:

- Bemenet: $N \in \mathbb{N}, X \in H^N$
- Kimenet: $Van \in L, Ind \in \mathbb{N}$
- 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

Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

Specifikáció:

Specifikáció:

- Bemenet: $N \in \mathbb{N}, X \in H^N$
- Kimenet: $Van \in L, Ind \in \mathbb{N}$
- 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!

➤ Bemenet: $N \in \mathbb{N}, X_{1..N} \in H^N$

$Y \in H$

➤ Kimenet: $Van \in L, Ind \in \mathbb{N}$

➤ Előfeltétel: **RendezettE(X)**

➤ Utófeltétel: $(Van, Ind) = \bigvee_{i=1}^N \text{Keres } i$
 $X_i = Y$

➤ Definíció (emlékeztető):

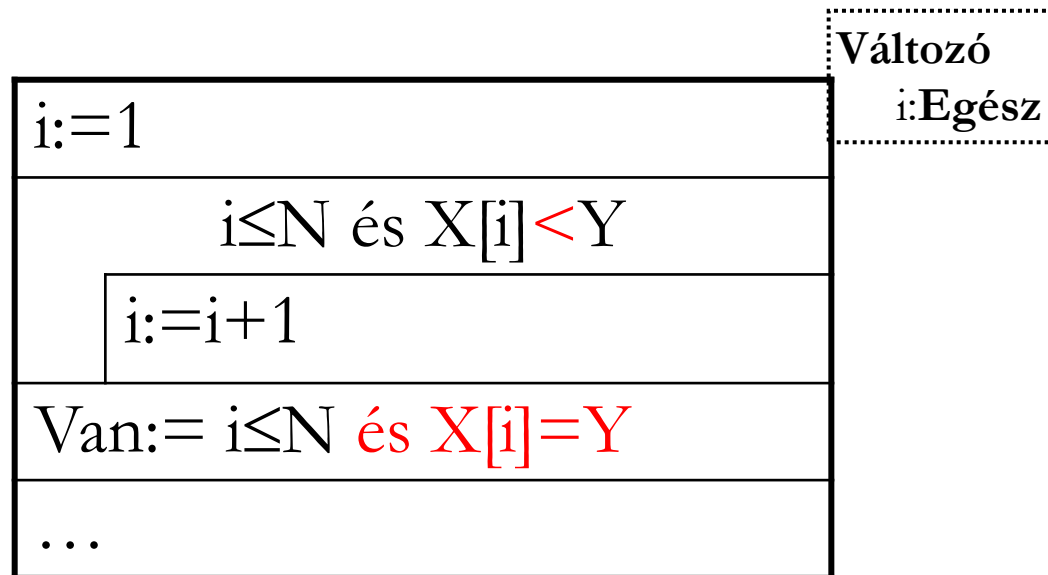
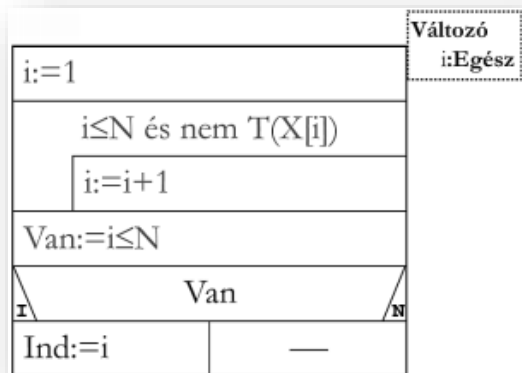
$\text{RendezettE}(X_{1..N}) := \forall i (1 \leq i < N): X_i \leq X_{i+1}$



Keresés rendezett sorozatban

Ötlet:

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



Észrevétel:

Van megoldás \leftrightarrow 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: $Van \in L$, $Ind \in \mathbb{N}$
- Előfeltétel: $N > 0$ és $RendezettE(X)$
- Utófeltétel: $Van = \exists i (1 \leq i \leq N) : X_i = Y$ 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

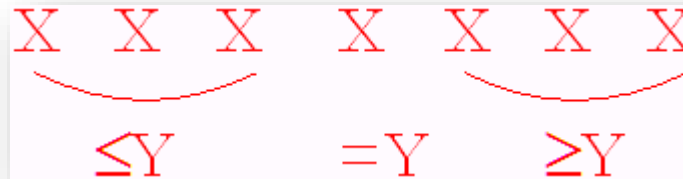
Van: **Logikai**, Ind: **Egész**

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

$e := 1$	
$u := N$	
$k := (e + u) \text{ Div } 2$	
$X[k] > Y$	$X[k] < Y$
$u := k - 1$	$e := k + 1$
$e \leq u \text{ és } X[k] \neq Y$	
$\text{Van} := X[k] = Y$	
...	

Specifikáció:

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



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



Rendezettek uniója

➤ Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^M 1$ és

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

➤ Utófeltétel₂: $(Db, Z) = \text{Unió}(N, X, M, Y)$ és RendezettE(Z)

Ötlet:

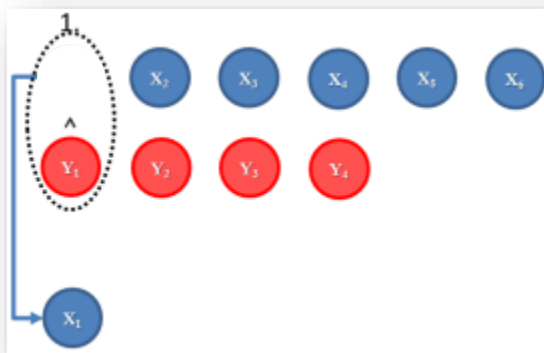
Az eredmény első eleme vagy az X, vagy az Y első eleme lehet. A kettő közül a rendezettség szerintit tegyük az eredménybe, majd a maradékra ugyanezt az elvet alkalmazhatjuk.



Rendezettek uniója

Algoritmus elé:

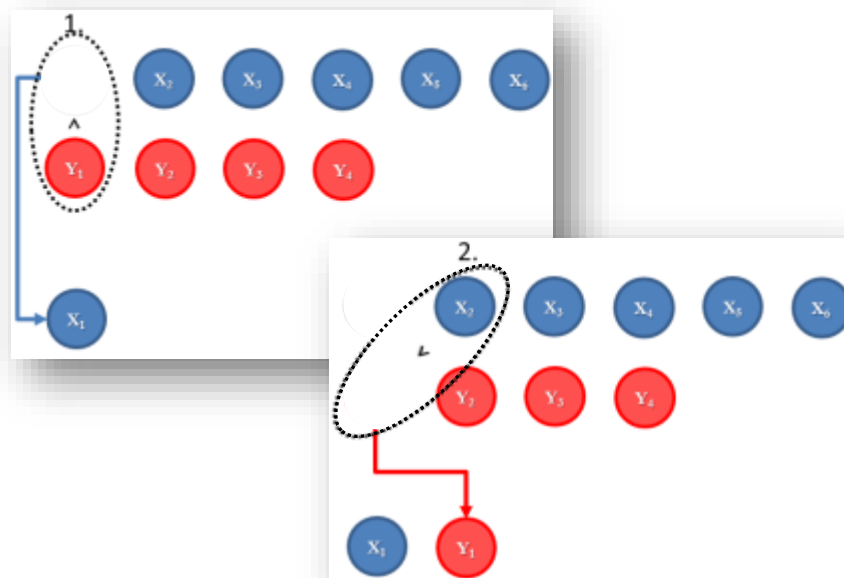
➤ Amíg van mit hasonlítani:



Rendezettek uniója

Algoritmus elé:

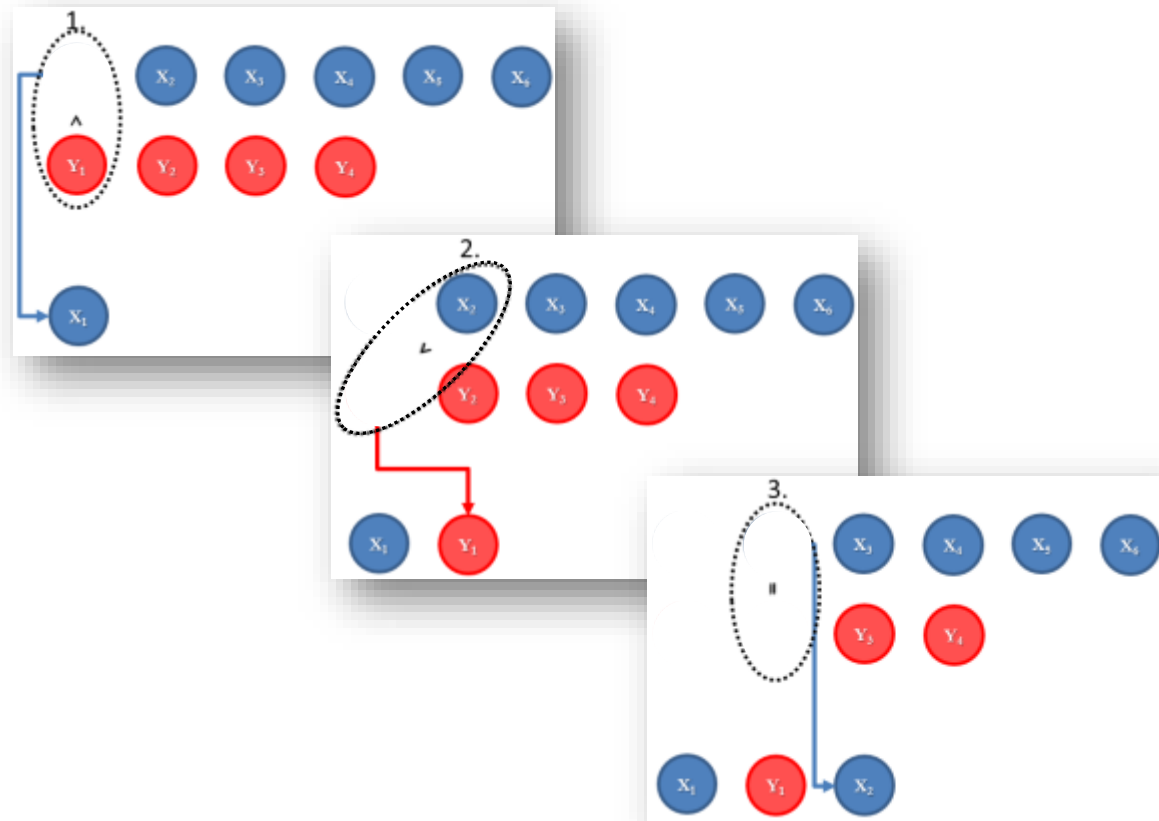
➤ Amíg van mit hasonlítani:



Rendezettek uniója

Algoritmus elé:

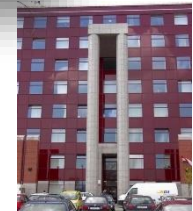
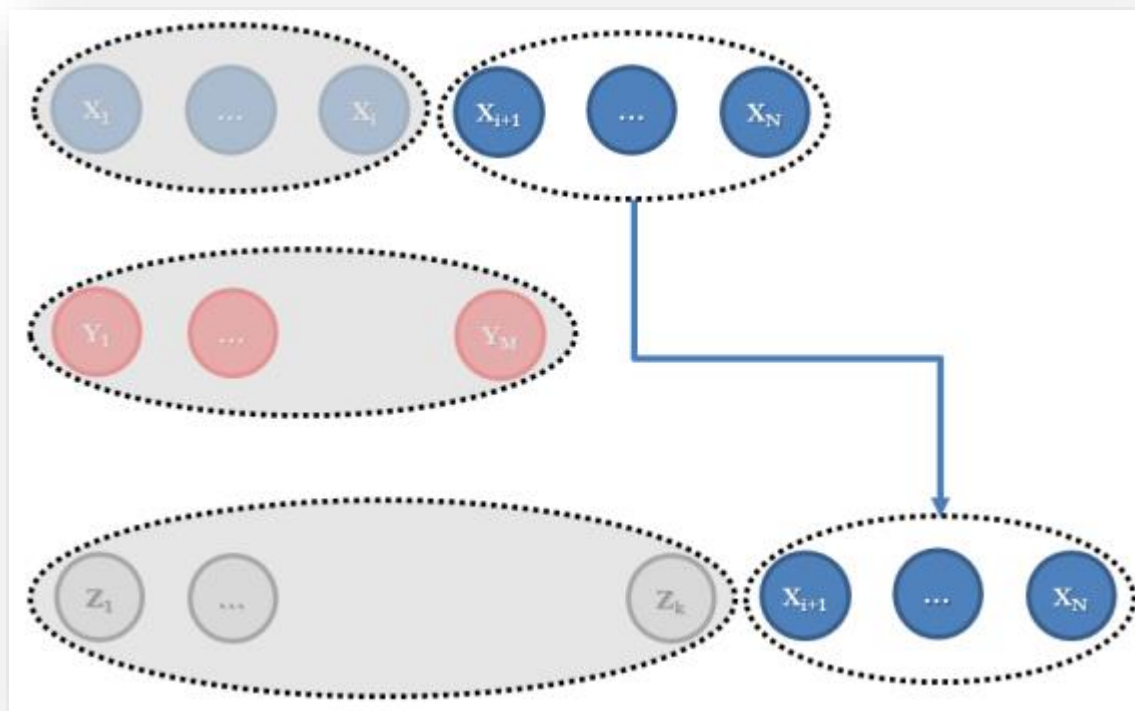
➤ Amíg van mit hasonlítani:



Rendezettek uniója

Algoritmus elé:

➤ Ha már nincs mit hasonlítani:



Rendezettek uniója

Algoritmus₁:

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{j=1}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y \text{ és } \text{HalmazE}(Z) \text{ és } \text{RendezettE}(Z)$

Változ

$i, j: \mathbb{Eg}$

$i := 1$

$j := 1$

$Db := 0$

$i \leq N$ és $j \leq M$

$Db := Db + 1$

$X[i] < Y[j]$

$X[i] = Y[j]$

$X[i] > Y[j]$

$Z[Db] := X[i]$

$Z[Db] := X[i]$

$Z[Db] := Y[j]$

$i := i + 1$

$i := i + 1$

$j := j + 1$

$j := j + 1$

...

Van miket hasonlítani

$Z := X$	
$Db := N$	
$j := 1..M$	
$i := 1$	
$i \leq N \text{ és } X[i] \neq Y[j]$	
$i := i + 1$	
$i > N$	
$Db := Db + 1$	—
$Z[Db] := Y[j]$	—



Rendezettek uniója

Algoritmus₁:

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y$ és $\text{HalmazE}(Z)$ és $\text{RendezettE}(Z)$

Változ

$i, j: \text{Eg}$

$i := 1$

$j := 1$

$Db := 0$

$i \leq N$ és $j \leq M$

$Db := Db + 1$

$X[i] < Y[j]$

$X[i] = Y[j]$

$Z[Db] := X[i]$

$Z[Db] := X[i]$

$Z[Db] := Y[j]$

$i := i + 1$

$i := i + 1$

$j := j + 1$

$j := j + 1$

...

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel: $Db = N + \sum_{j=1}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y$ és $\text{HalmazE}(Z)$ és $\text{RendezettE}(Z)$

Nincs Y-beli.

Nincs X-beli.

$Z := X$
$Db := N$
$j = 1..M$
$i := 1$
$i \leq N \text{ és } X[i] \neq Y[j]$
$i := i + 1$
$i > N$
$Db := Db + 1$
$Z[Db] := Y[j]$

\dots
$i \leq N$
$Db := Db + 1$
$Z[Db] := X[i]$
$i := i + 1$
$j \leq M$
$Db := Db + 1$
$Z[Db] := Y[j]$
$j := j + 1$



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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel: $Db = N + \sum_{j=1}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y$ és $\text{HalmazE}(Z)$ és $\text{RendezettE}(Z)$

Nincs Y-beli.

Nincs X-beli.

...		
$i \leq N$		
Db := Db + 1		
Z[Db] := X[i]		
i := i + 1		
$j \leq M$		
Db := Db + 1		
Z[Db] := Y[j]		
j := j + 1		

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

$i \leq N$ és $j \leq M$		
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	



Rendezettek uniója

Algoritmus₂:

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y \text{ és } \text{HalmazE}(Z) \text{ és } \text{RendezettE}(Z)$

Változ
 $i, j: \mathbb{E}$

$i:=1$		
$j:=1$		
$Db:=0$		
$X[N+1]:=+\infty$		
$Y[M+1]:=+\infty$		
$i \leq N+1$ és $j \leq M+1$		
$Db:=Db+1$		
$X[i] < Y[j]$	$X[i] = Y[j]$	$X[i] > Y[j]$
$Z[Db]:=X[i]$	$Z[Db]:=X[i]$	$Z[Db]:=Y[j]$
$i:=i+1$	$i:=i+1$	$j:=j+1$
	$j:=j+1$	

... és utoljára?
 $Z[Db]:=+\infty$



Rendezettek uniója

Algoritmus₂:

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y \text{ és } \text{HalmazE}(Z) \text{ és } \text{RendezettE}(Z)$

Változ
 $i, j: \mathbb{E}$

$i := 1$

$j := 1$

$Db := 0$

$X[N+1] := +\infty$

$Y[M+1] := +\infty$

$i \leq N+1$ és $j \leq M+1$

$Db := Db + 1$

$X[i] < Y[j]$

$X[i] = Y[j]$

$Z[Db] := X[i]$

$Z[Db] := X[i]$

$Z[Db] := Y[j]$

$i := i + 1$

$i := i + 1$

$j := j + 1$

$j := j + 1$

... és utoljára?
 $Z[Db] := +\infty$



Rendezettek uniója

Algoritmus₂ 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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^M 1$ és
 $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y \text{ és}$
 $\text{HalmazE}(Z) \text{ és } \text{RendezettE}(Z)$

Változ
i,j:E

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

$i:=1$												
$j:=1$												
$Db:=0$												
$X[N+1]:=+\infty$												
$Y[M+1]:=+\infty$												
$i < N+1$ vagy $j < M+1$												
$Db:=Db+1$												
<table><tr><td>$X[i] < Y[j]$</td><td>$X[i] = Y[j]$</td><td></td></tr><tr><td>$Z[Db]:=X[i]$</td><td>$Z[Db]:=X[i]$</td><td>$Z[Db]:=Y[j]$</td></tr><tr><td>$i:=i+1$</td><td>$i:=i+1$</td><td>$j:=j+1$</td></tr><tr><td></td><td>$j:=j+1$</td><td></td></tr></table>	$X[i] < Y[j]$	$X[i] = Y[j]$		$Z[Db]:=X[i]$	$Z[Db]:=X[i]$	$Z[Db]:=Y[j]$	$i:=i+1$	$i:=i+1$	$j:=j+1$		$j:=j+1$	
$X[i] < Y[j]$	$X[i] = Y[j]$											
$Z[Db]:=X[i]$	$Z[Db]:=X[i]$	$Z[Db]:=Y[j]$										
$i:=i+1$	$i:=i+1$	$j:=j+1$										
	$j:=j+1$											



Rendezettek uniója

Algoritmus₂ javítása:

Változ
i,j:E

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: $\text{HalmazE}(X)$ és $\text{HalmazE}(Y)$ és $\text{RendezettE}(X)$ és $\text{RendezettE}(Y)$
- Utófeltétel₁: $Db = N + \sum_{\substack{j=1 \\ Y_j \in X}}^M 1$ és $\forall i (1 \leq i \leq Db): Z_i \in X \text{ vagy } Z_i \in Y \text{ és } \text{HalmazE}(Z) \text{ és } \text{RendezettE}(Z)$

i:=1

j:=1

Db:=0

$X[N+1] := +\infty$

$Y[M+1] := +\infty$

$i \leq N$ vagy $j \leq M$

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:=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	



Rendezettek uniója

Kérdések:

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

Tapasztalat:

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



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

Rendezettek összefésülése



Feladat:

Adott két rendezett sorozat, adjuk meg az összefésülésüket!

Specifikáció:

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



Rendezettek összefésülése



- Utófeltétel: $Z \in \text{Permutáció}(X \oplus Y)$ és
 $\text{Rendezett}E(Z)$

Ötlet:

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



Rendezettek összefésülése



Algoritmus:

Változó
 i, j : Egész

i:=1	
j:=1	
Db:=0	
X[N+1]:=+∞	
Y[M+1]:=+∞	
i≤N vagy j≤M	
Db:=Db+1	
<div><div>i</div><div>X[i] ≤ Y[j]</div><div>j</div></div>	<div><div>X[i] ≤ Y[j]</div><div>j</div></div>
Z[Db]:=X[i]	Z[Db]:=Y[j]
i:=i+1	j:=j+1

Specifikáció:

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

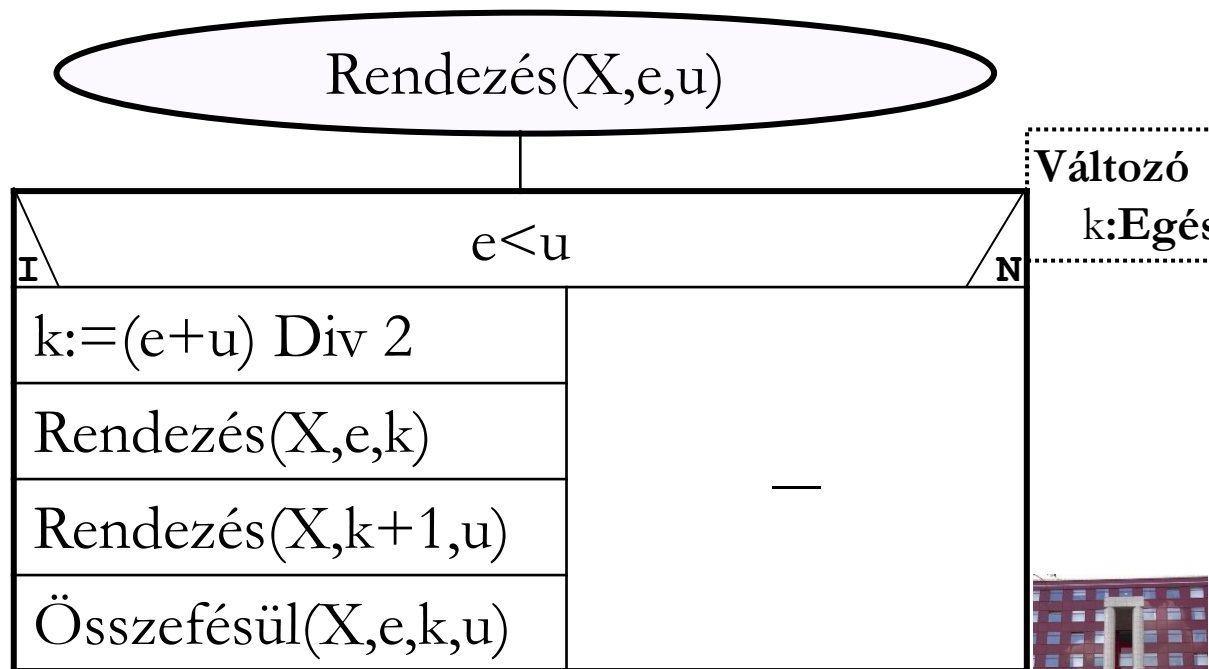


Összefésüléses rendezés



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 részfadatainak 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:

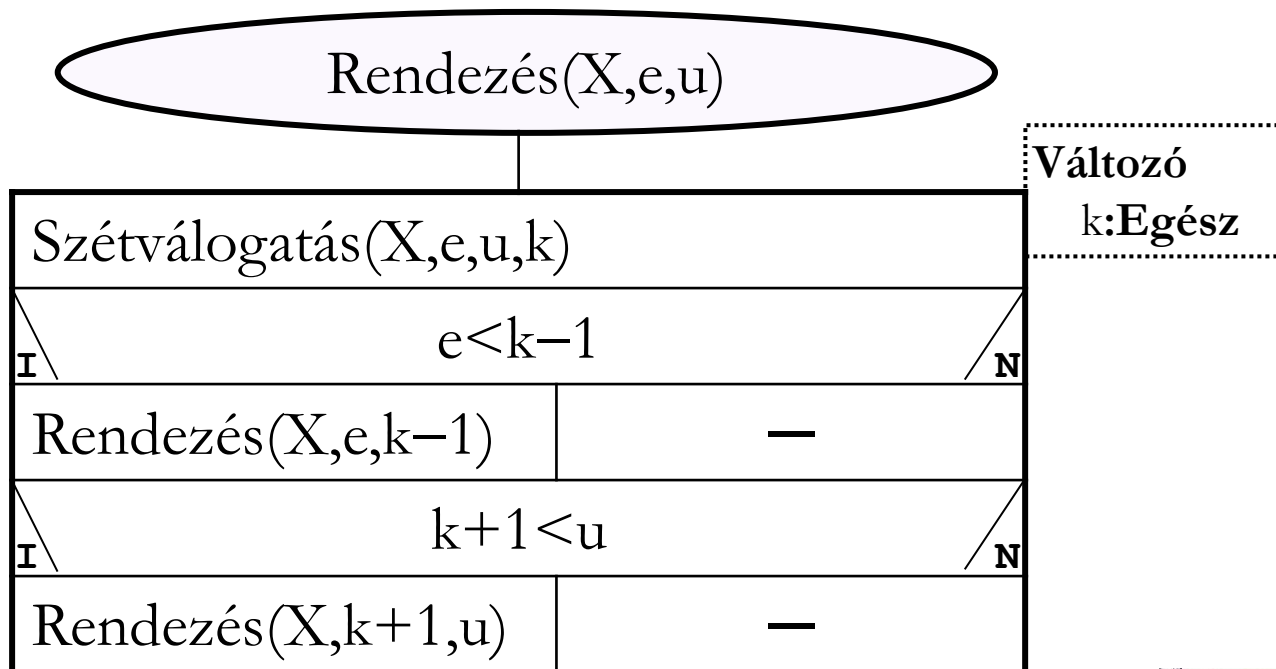
Gyorsrendezés (quicksort):

- felbontás: $\boxed{X_1, \dots, X_{k-1}} \ X_k \ \boxed{X_{k+1}, \dots, X_n}$ szétválogatás
ahol $\forall i, j \ (1 \leq i < k; k < j \leq n): X_i \leq X_k \text{ és } X_k \leq X_j$
- 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



➤ Rendezési feladat

- Specifikáció
- Egyszerű cserés rendezés
- Minimum-kiválasztásos rendezés
- Buborékos rendezés
- Javított buborékos rendezés

- Beillesztéses rendezés
- Javított beillesztéses rendezés
- Szétoztó rendezés
- Számlálva szétoztó rendezés
- Számláló rendezés

➤ Rendezések hatékonysága – idő

➤ Algoritmusok rendezett sorozatokban

- Keresés rendezett sorozatban
- Rendezettek uniója, összefésülése
- Összefésüléssel rendezés

➤ Oszd meg és uralkodj!

