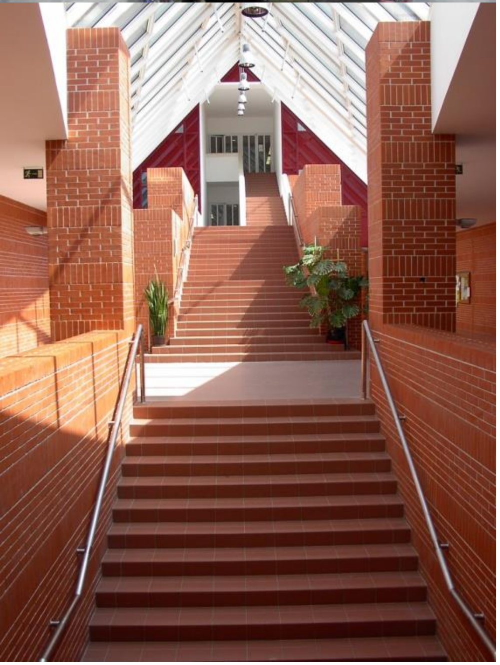
Programozási alapismeretek
12. előadás

Programozási alapismeretek
12. előadás



- Tapasztalatok a rendezésről
- Keresés rendezett sorozatban
- Rendezettek uniója
- Rendezettek összefésülése
- Kiválogatás helyben
- Szétválogatás helyben
- Hatékonyságvizsgálat táblázatkezelővel

Tapasztalatok a rendezésről

- A rendezési algoritmusok eredménye egy rendezett sorozat. Vajon lehet-e a korábban megismert feladatokat haté-konyabban megoldani, ha a bemenetük rendezett?
- A rendezési algoritmusok többsége „helyben” rendez. Vannak-e más, „helyben” működő algoritmusok?

Keresés rendezett sorozatban

Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

T-tulajdonság:
$T(x) := (x = Y)$

Specifikáció:

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

Konkretizáljuk:
legyen növekvő!

Keresés rendezett sorozatban

Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

Specifikáció:

T-tulajdonság: $T(x):=(x=Y)$

- Bemenet: $N \in \mathbb{N}$, $X \in \mathbb{H}^N$
  $Y \in \mathbb{H}$
- Kimenet: $Van \in \mathbb{L}$, $Ind \in \mathbb{N}$
- Előfeltétel: RendezettE(X)

Konkretizáljuk: legyen növekvő!

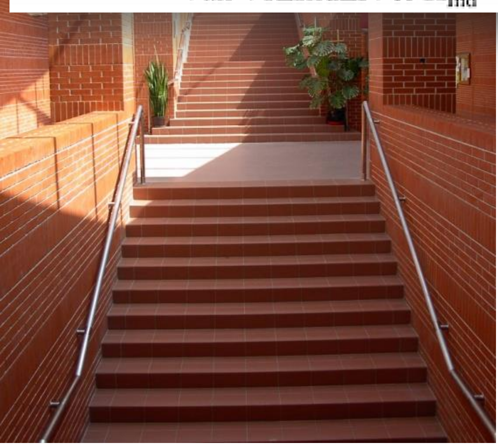
- Utófeltétel: $(Van,Ind)= \underset{\substack{i=1 \\ X_i=Y}}{\overset{N}{\text{Keres}}}\ i$
- Definíció (emlékeztető):
  $RendezettE(X_{1..N}):=\forall i(1\leq i<N): X_i \leq X_{i+1}$

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

Változó
i:Egész

i:=1	
i≤N és nem T(X[i])	
i:=i+1	
Van:=i≤N	
Van (I / N)	
Ind:=i	—

Specifikáció:
- Bemenet: $N \in \mathbb{N}$, $Y \in H$, $X \in H^N$
- 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 $X_{Ind} = Y$



Keresés rendezett sorozatban

Ötlet:

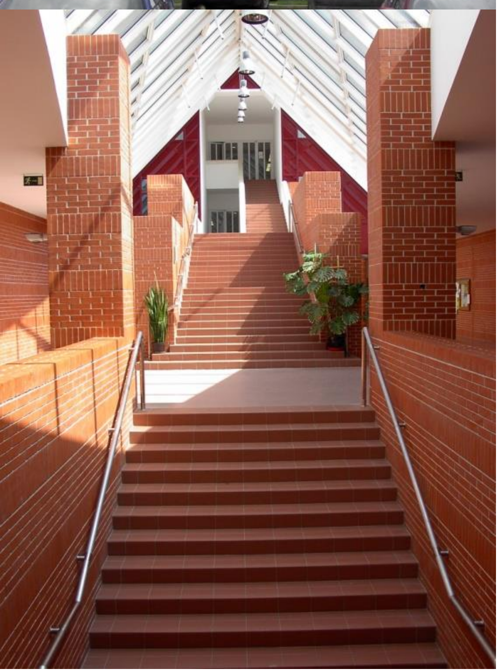
Ha már a keresett elem értékénél nagyobbnál tartunk, akkor biztos nem lesz a sorozatban, megállhatunk.

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

Programparaméterek:

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

Specifikáció:

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

Ötlet és –tömb esetén– lehetőség:

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

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

Algoritmus:

X X X X X X X

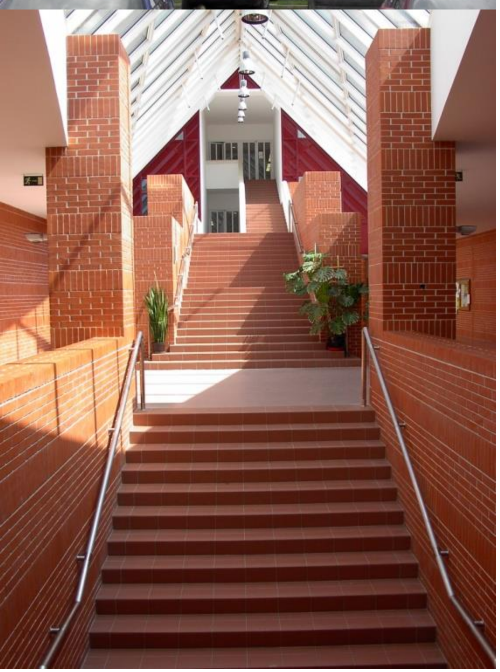
$\underbrace{\text{X X X}}_{\le Y}$ $=Y$ $\underbrace{\text{X X X}}_{\ge Y}$

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
...
```

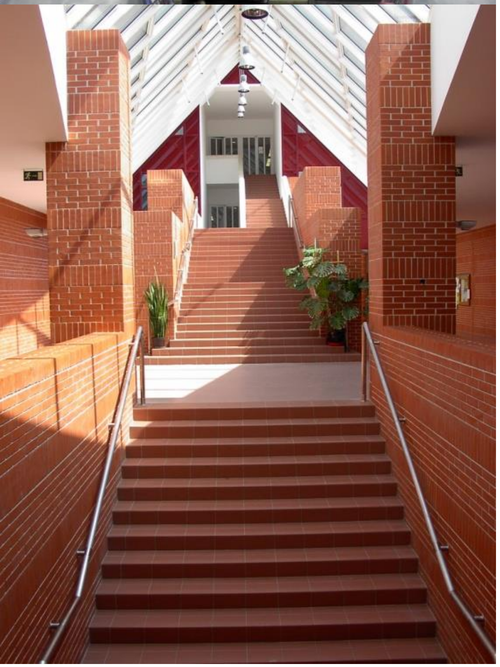
Itt akkor van megoldás, ha megtaláltuk a keresett érték valamelyikét.

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.

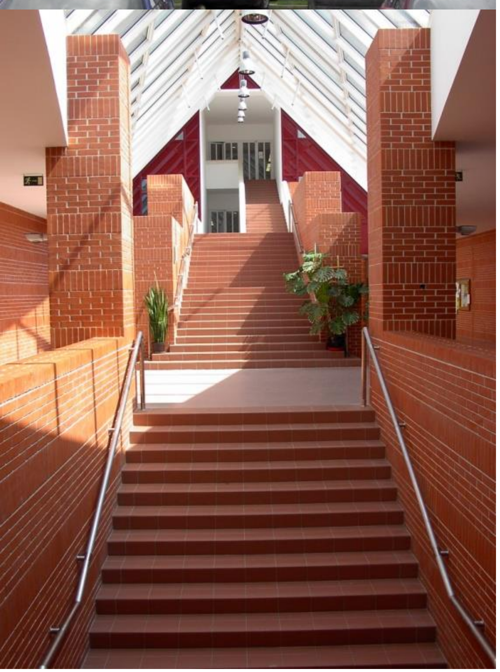
Feladat:

Adott két rendezett halmaz, adjuk meg az uniójukat!

Specifikáció:

- Bemenet: $N,M \in \mathbb{N}$, $X \in \mathbb{H}^N$, $Y \in \mathbb{H}^M$
- Kimenet: $Db \in \mathbb{N}$, $Z \in \mathbb{H}^{Db}$

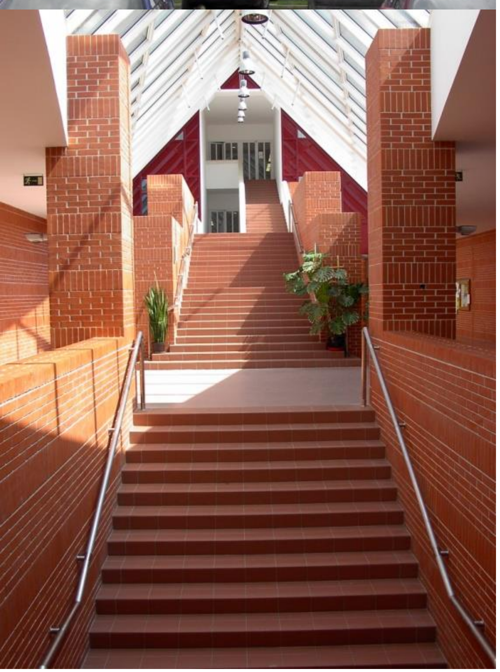
  $\leq N+M$
- Előfeltétel: HalmazE(X) és HalmazE(Y) és RendezettE(X) és RendezettE(Y)



Rendezettek uniója

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

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



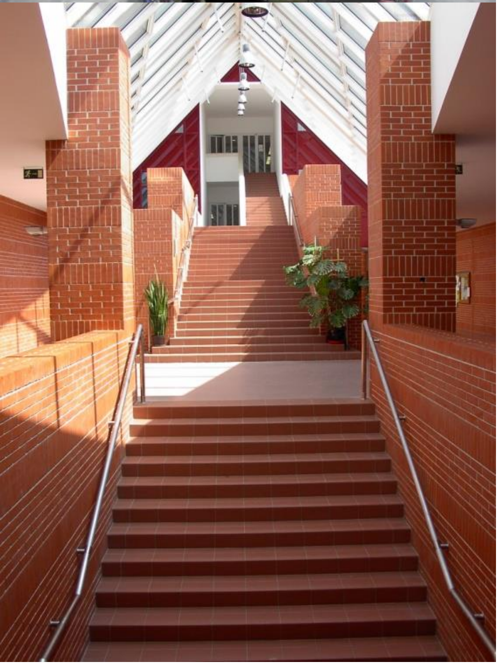
Rendezettek uniója

- 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)

- Definíció (emlékeztető):

  $\text{HalmazE}(X_{1..N}) :\!= \forall i \neq j \in [1..N]: X_i \neq X_j$



Rendezettek uniója

- Utófeltétel$_2$: $(Db,Z)=\text{Unió}(N,X,M,Y)$ és RendezettE(Z)

- Definíció (emlékeztető):

  $\text{HalmazE}(X_{1..N}) := \forall i \neq j \in [1..N]: X_i \neq X_j$

Rendezettek uniója

Csak tömb esetén működhet?

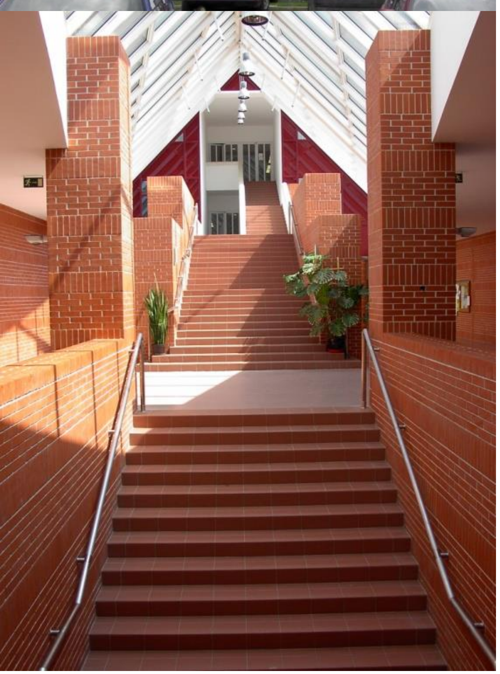
- Utófeltétel$_2$: (Db,Z)=Unió(N,X,M,Y) és RendezettE(Z)

- Definíció (emlékeztető):

  $\text{HalmazE}(X_{1..N}) := \forall i \neq j \in [1..N]: X_i \neq X_j$

Ötlet:

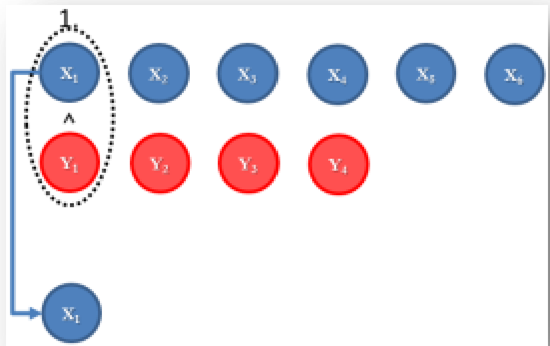
Az eredmény **első** eleme vagy az X, vagy az Y első eleme lehet. A kettő közül a **rendezett-ség szerintit** tegyük az eredménybe, majd a maradékra ugyanezt az elvet alkalmazhatjuk.

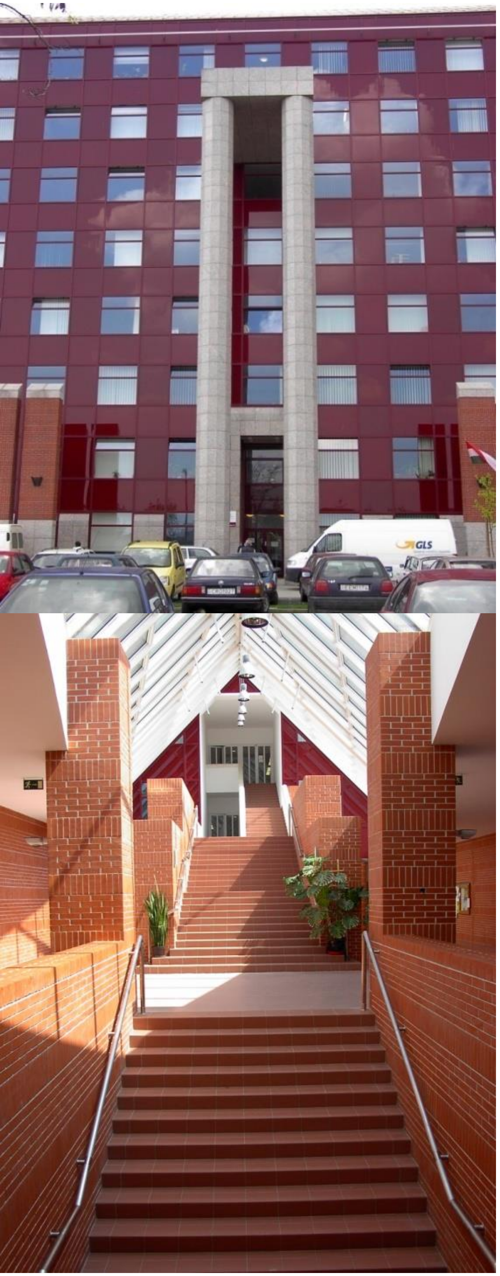


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

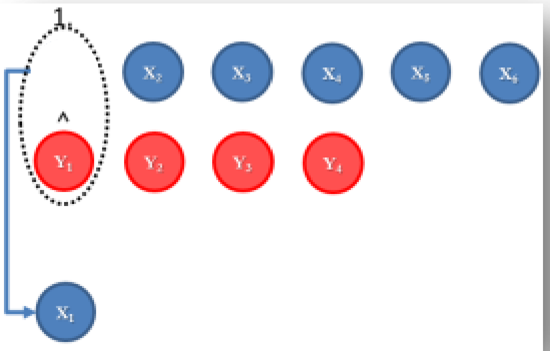


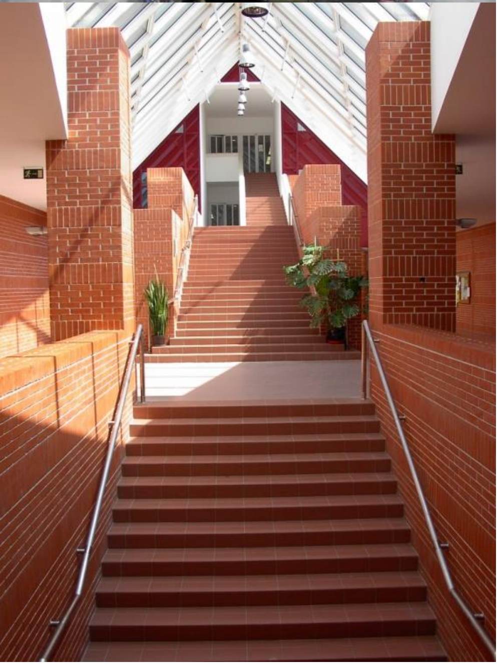


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

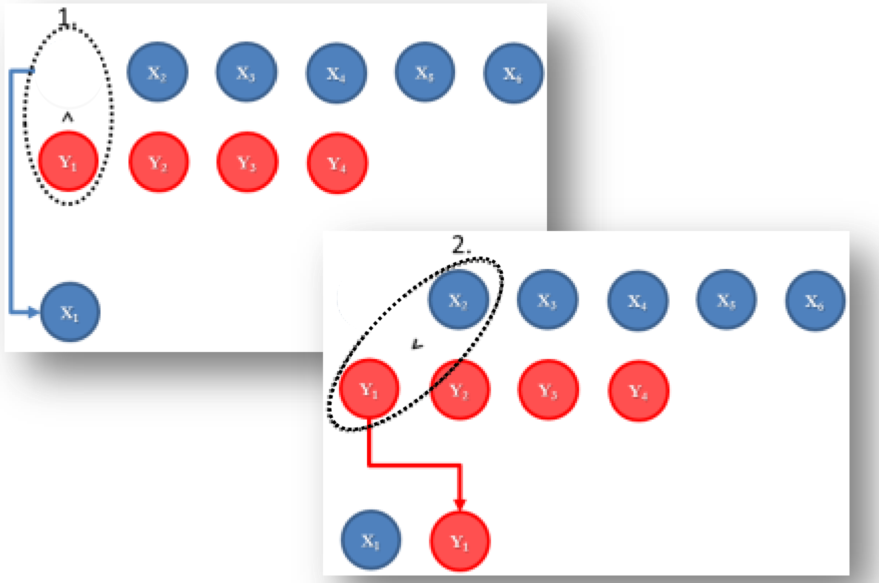


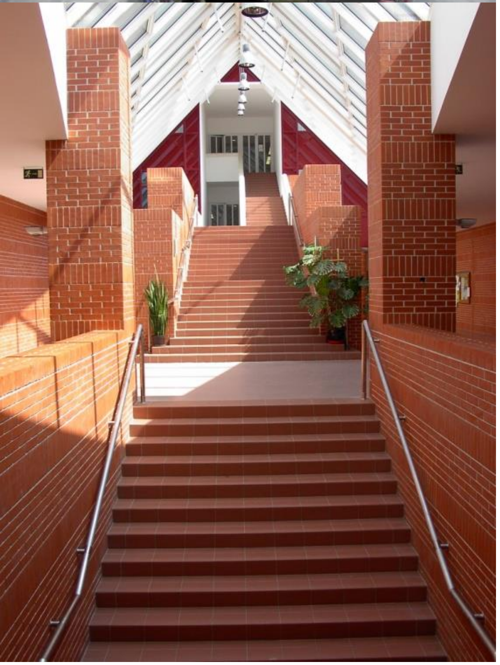


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

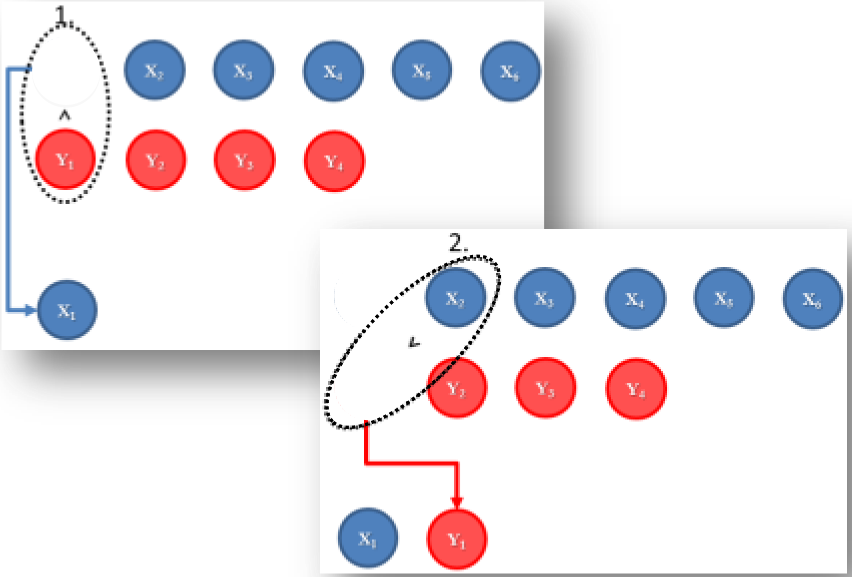


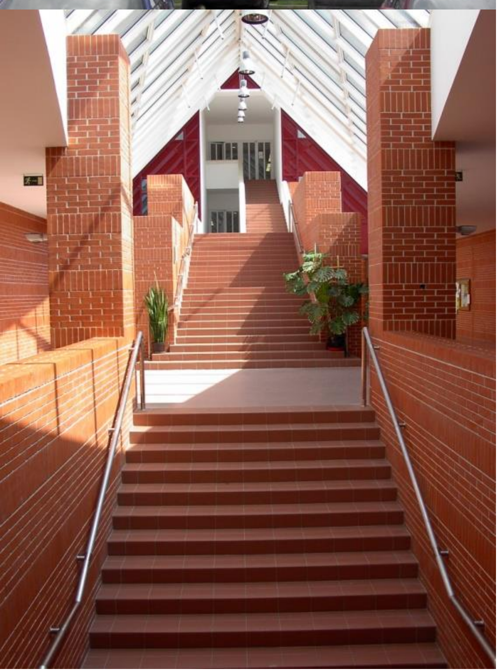


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

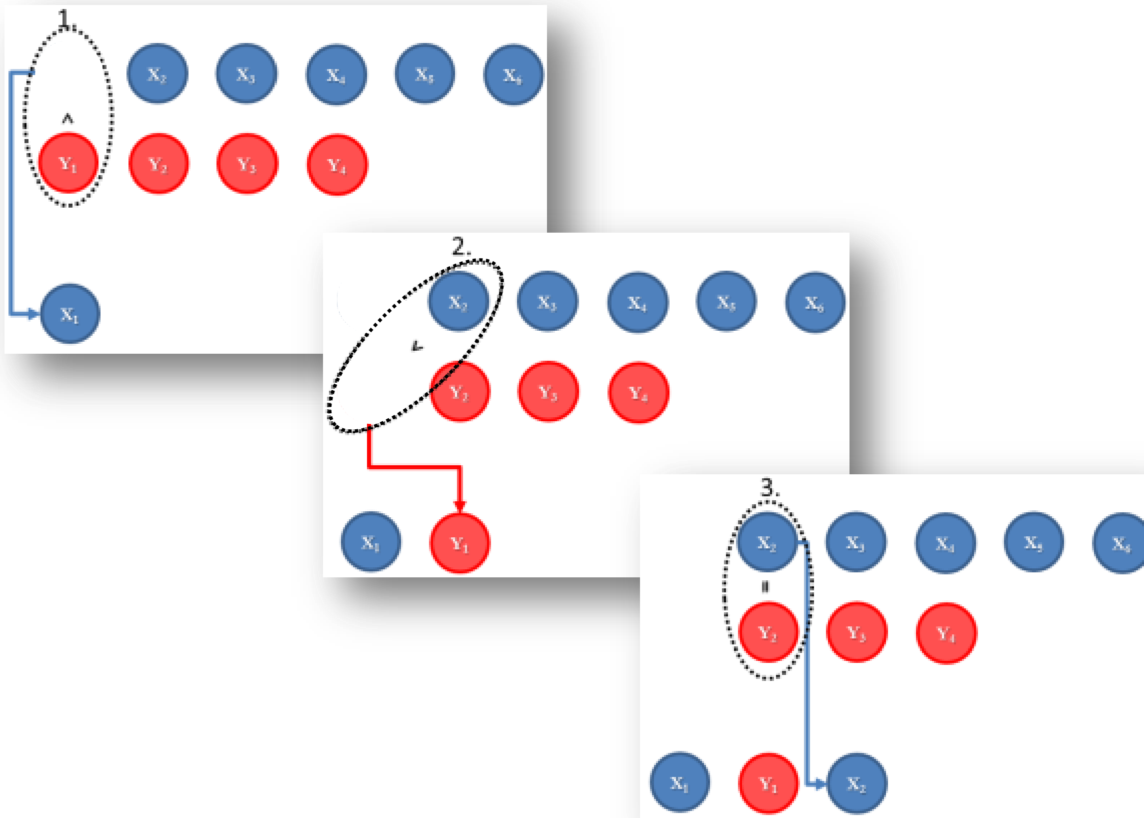


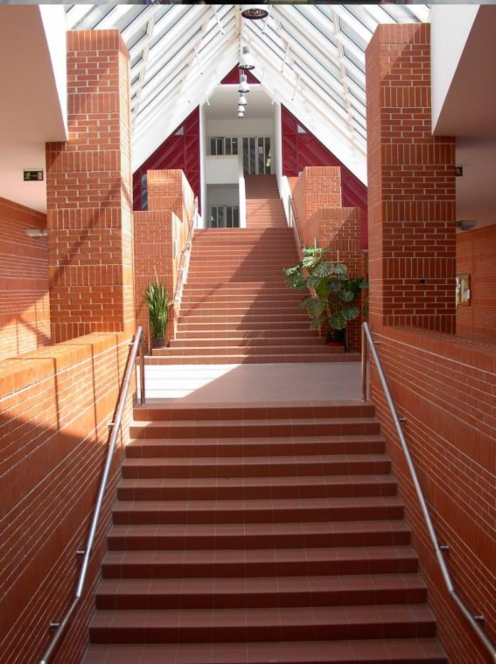


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

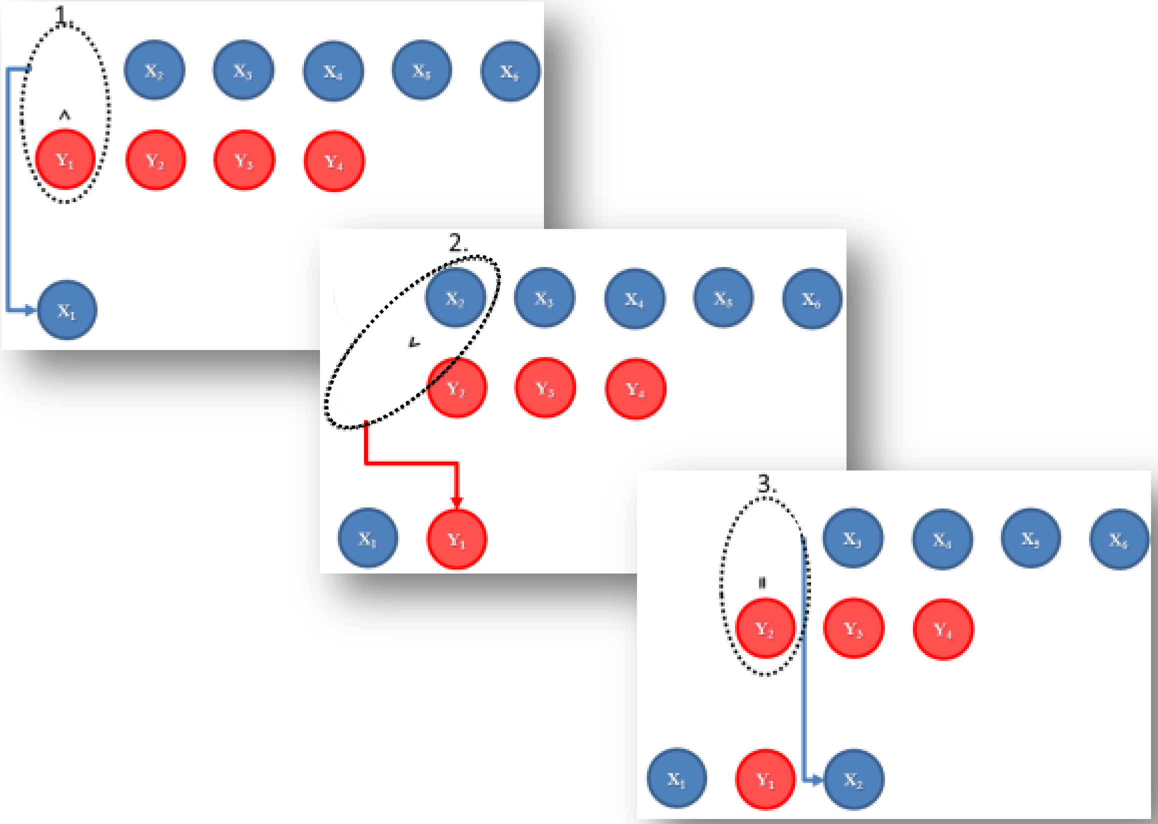


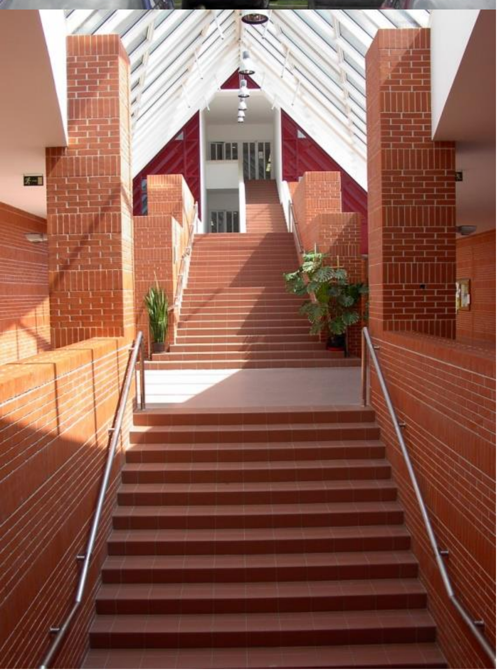


Rendezettek uniója

Algoritmus elé:

- Amíg van mit hasonlítani:

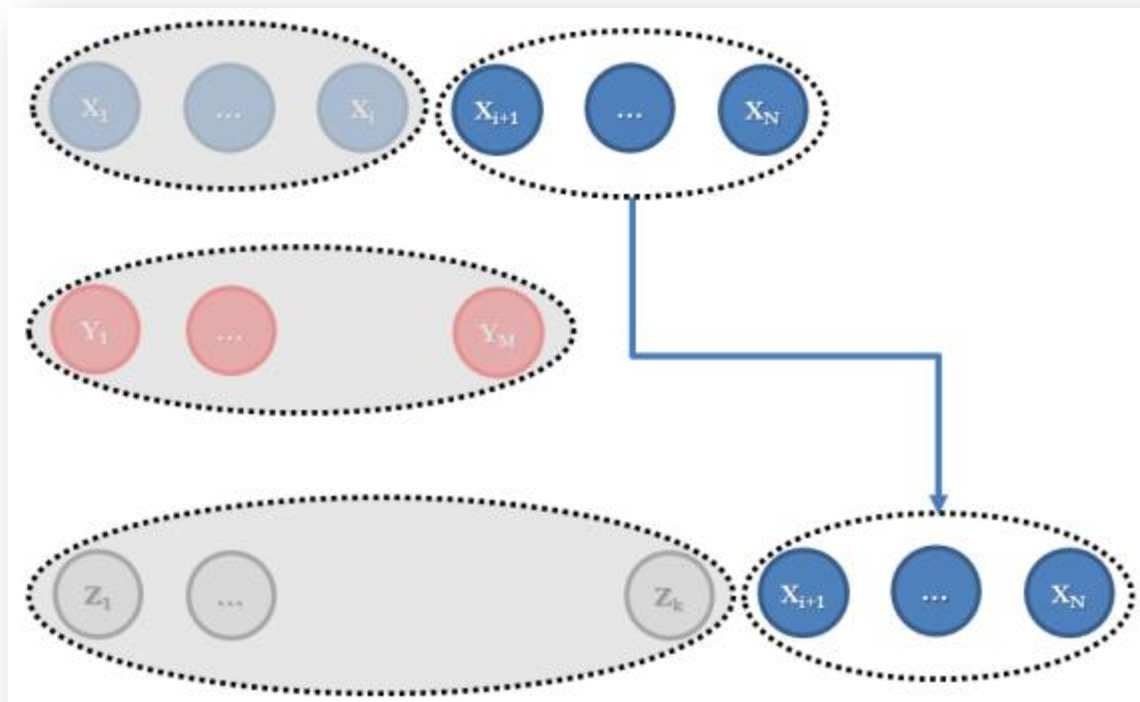




Rendezettek uniója

Algoritmus elé:

- Ha már nincs mit hasonlítani:



Rendezettek uniója

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)

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]

Algoritmus$_1$:

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	
…		

Változó
i,j:Eg

Rendezettek uniója

Specifikáció:

- Bemenet: $N,M\in N$, $X\in H^N$, $Y\in H^M$
- Kimenet: $Db\in N$, $Z\in H^{Db}$
- 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 $\forall i(1\leq i\leq Db)$: $Z_i\in X$ 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]

Algoritmus$_1$:

Változ

i,j:Eg

i:=1

j:=1

Db:=0

i≤N és j≤M

Van miket hasonlítani

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	

...

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)

Rendezettek uniója

…

i≤N

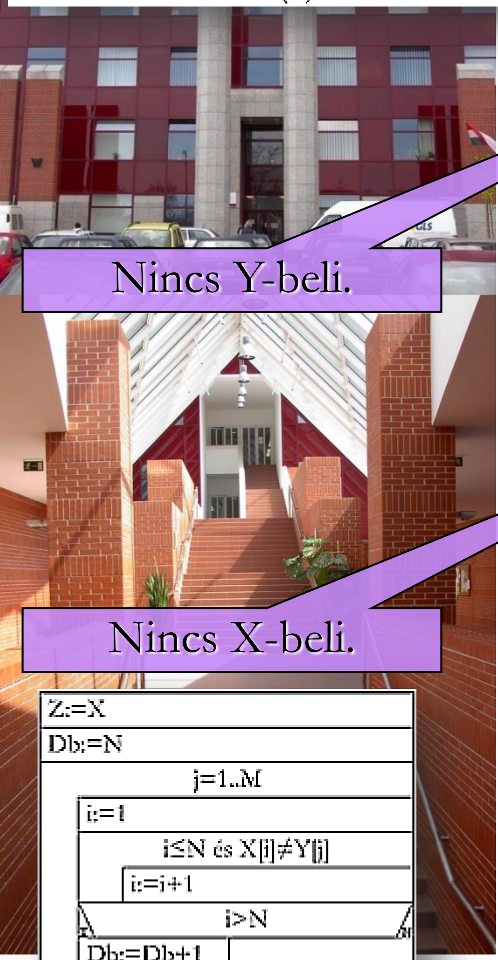
- Db:=Db+1
- Z[Db]:=X[i]
- i:=i+1

j≤M

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

Nincs Y-beli.

Nincs X-beli.



Z:=X

Db:=N

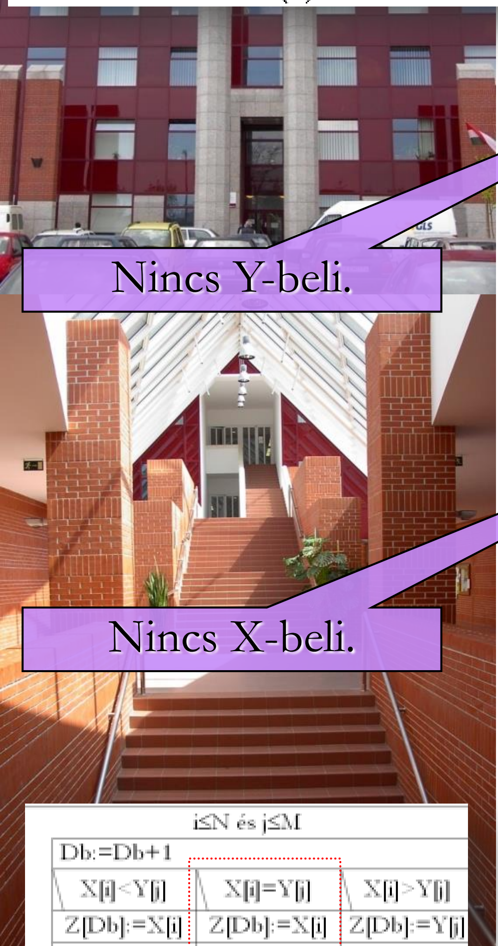
j=1..M

- i:=1
- i≤N és X[i]≠Y[j]
  - i:=i+1
- i>N
  - I: Db:=Db+1; Z[Db]:=Y[j]
  - N: —

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.

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 kellene!

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)

Rendezettek uniója

Algoritmus$_2$:

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	

Változ…
i,j:E…


… és utoljára?
Z[Db]:=+ ∞

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)

Rendezettek uniója

Algoritmus$_2$:

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

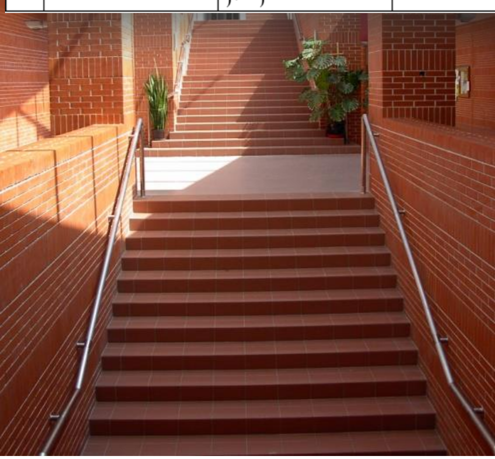
… és utoljára?
Z[Db]:=+ ∞

Rendezettek uniója

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	



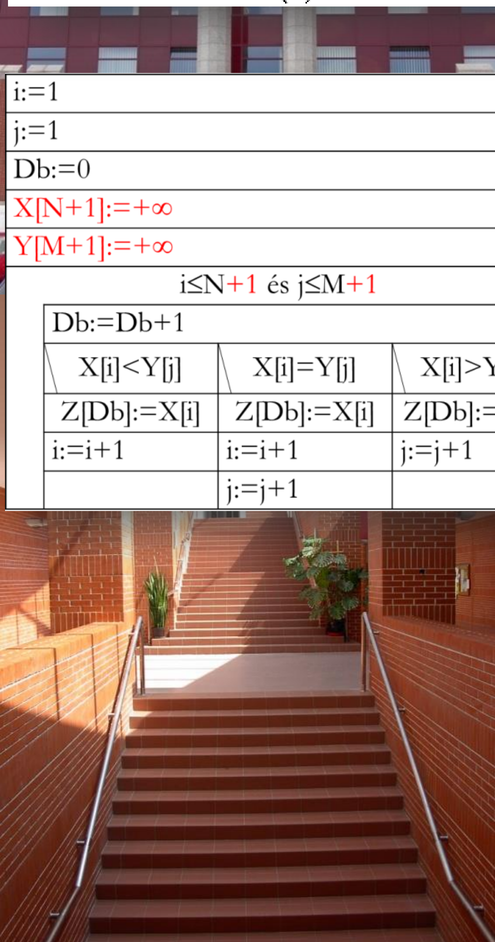
Algoritmus$_2$ javítása:

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

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	

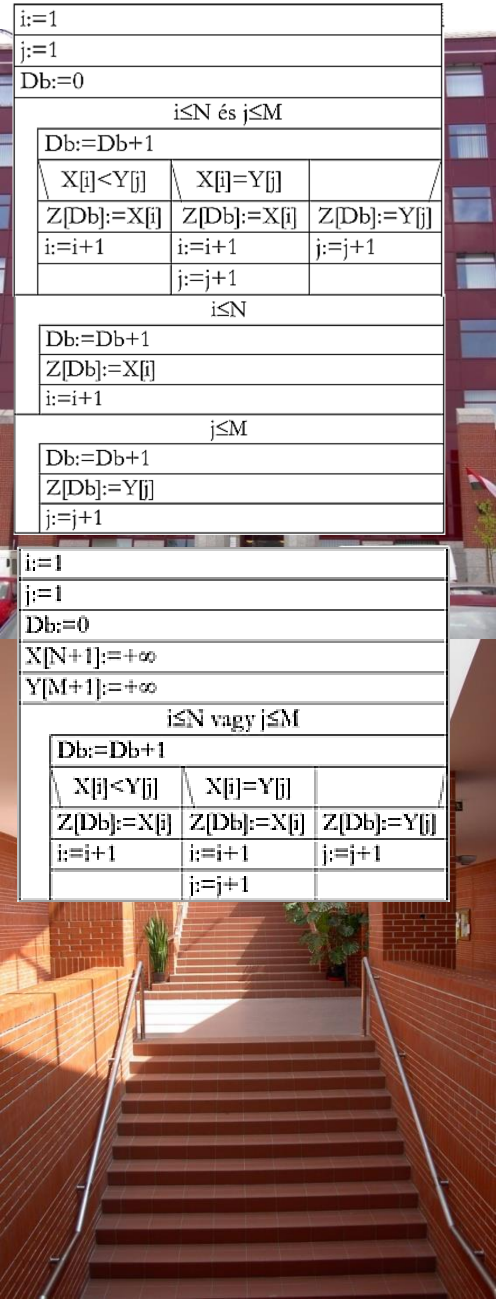
Rendezettek uniója

Algoritmus$_2$ javítása:

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

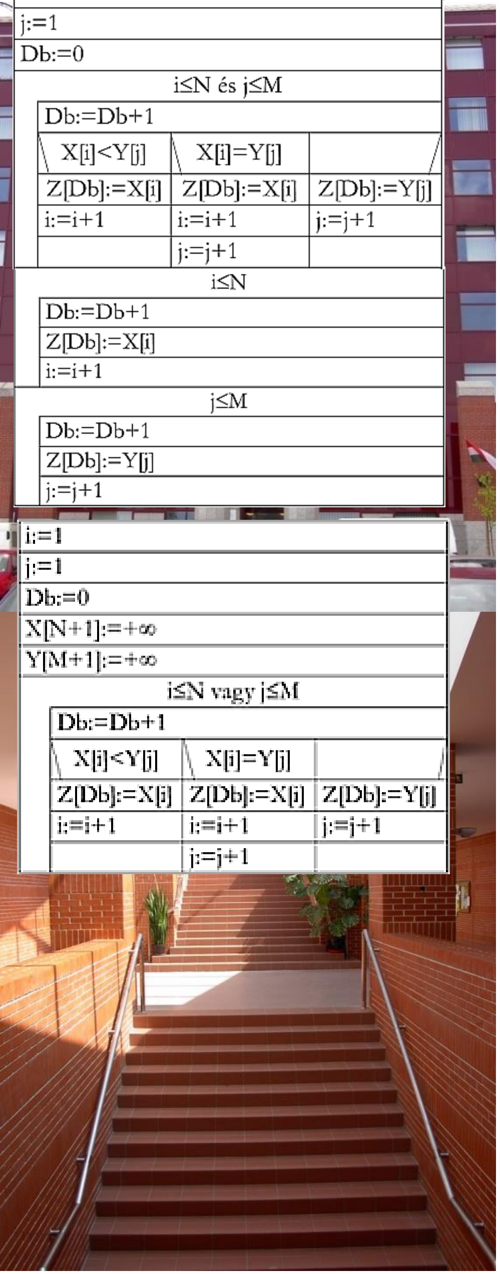


Rendezettek uniója

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?

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
```

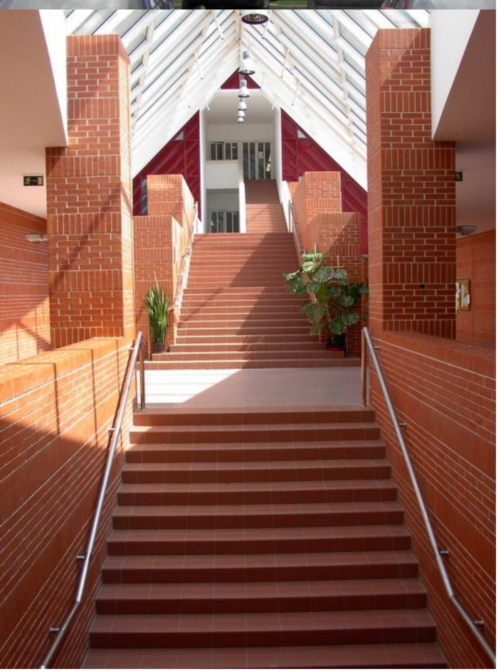
```
i:=1
j:=1
Db:=0
X[N+1]:=+∞
Y[M+1]:=+∞
i≤N vagy j≤M
    Db:=Db+1
    X[i]<Y[j]    | X[i]=Y[j]    |
    Z[Db]:=X[i]  | Z[Db]:=X[i]  | Z[Db]:=Y[j]
    i:=i+1       | i:=i+1       | j:=j+1
                 | j:=j+1       |
```

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 szem-pontjából. (⇐ Ciklus-/elágazás-szám.)
- Ez a módszer a kimenet szerint halad egye-sével és nem a bemenet szerint (mint a ko-rábbiak).



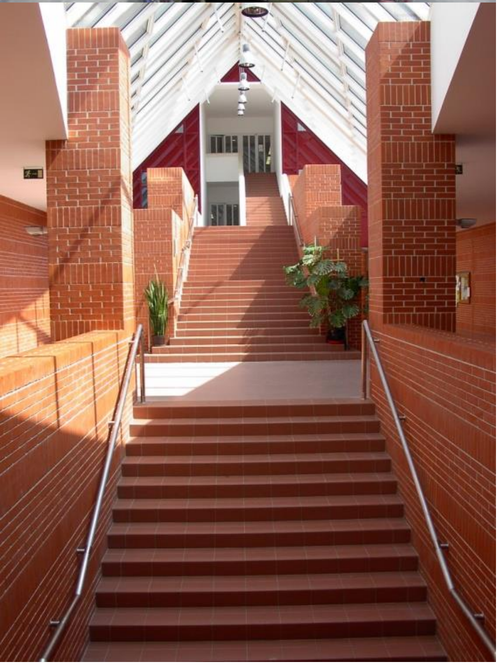
Rendezettek összefésülése

Feladat:

Adott két rendezett sorozat, adjuk meg az összefésülésüket!

Specifikáció:

- Bemenet: $N,M \in \mathbb{N}$, $X \in \mathbb{H}^N$, $Y \in \mathbb{H}^M$
- Kimenet: $Z \in \mathbb{H}^{N+M}$
- 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

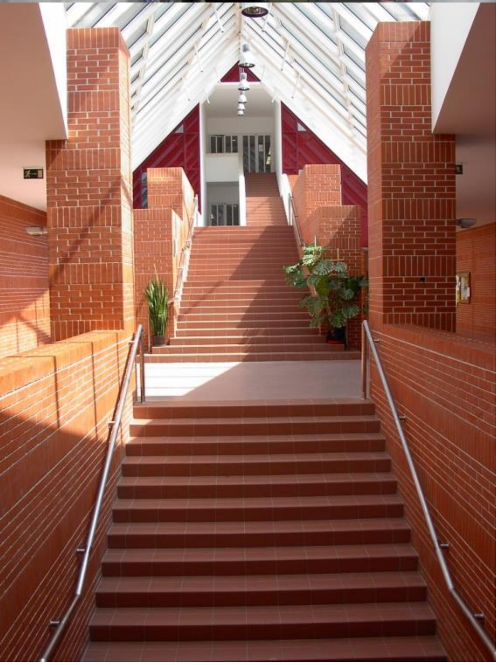
Specifikáció:

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

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



Kiválogatás helyben

Specifikáció:

- Bemenet: $N \in \mathbb{N},\ X \in \mathbb{H}^N$
- Kimenet: $Db \in \mathbb{N},\ X' \in \mathbb{H}^N$
- Előfeltétel: –
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ és

  $X'_{1..Db} \subseteq X$ és

  $\forall i(1 \le i \le Db): T(X'_i)$

Kiválogatás helyben

Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
- Kimenet: $Db \in \mathbb{N}$, $X' \in H^N$
- Előfeltétel: –
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ és

  $X'_{1..Db} \subseteq X$ és

  $\forall i(1 \leq i \leq Db): T(X'_i)$

Ötlet:

Itt olyan helyre tesszük a kiválogatott elemet, amelyre már nincs szükségünk.

Algoritmus:

Változó
i:**Egész**

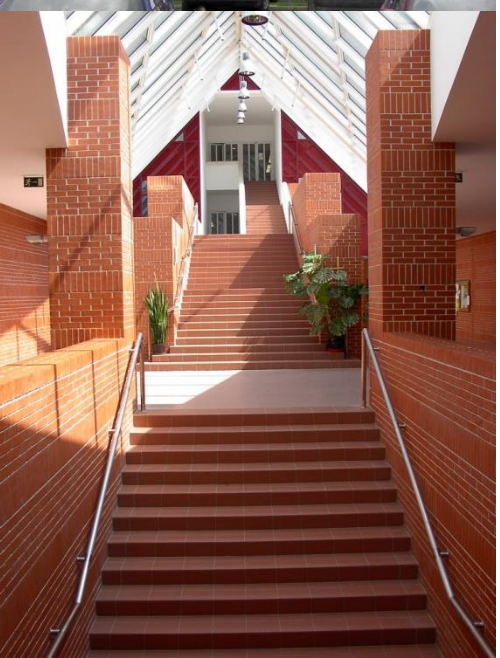
Db:=0

i=1..N

T(X[i])

I: Db:=Db+1; X[Db]:=X[i]

N: —



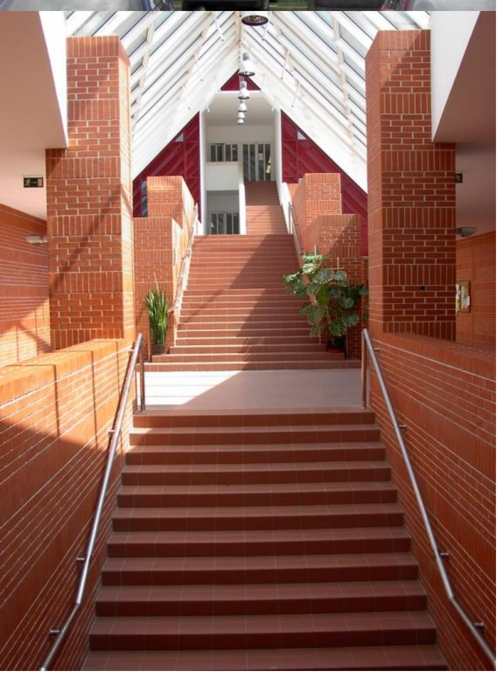
Szétválogatás helyben

Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X \in \mathbb{H}^N$
- Kimenet: $Db \in \mathbb{N}$, $X' \in \mathbb{H}^N$
- Előfeltétel: –
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ és $X' \in Permutáció(X)$

  és $\forall i(1 \leq i \leq Db)$: $T(X'_i)$
  és $\forall i(Db+1 \leq i \leq N)$: nem $T(X'_i)$
- Definíció (emlékeztető):

  Permutáció(Z):= Z elemeinek összes permutációjának halmaza

Szétválogatás helyben

Algoritmikus ötlet:

1. Vegyük ki (másoljuk le) a sorozat **első** elemét:

   O x x x x x x x x x x x x

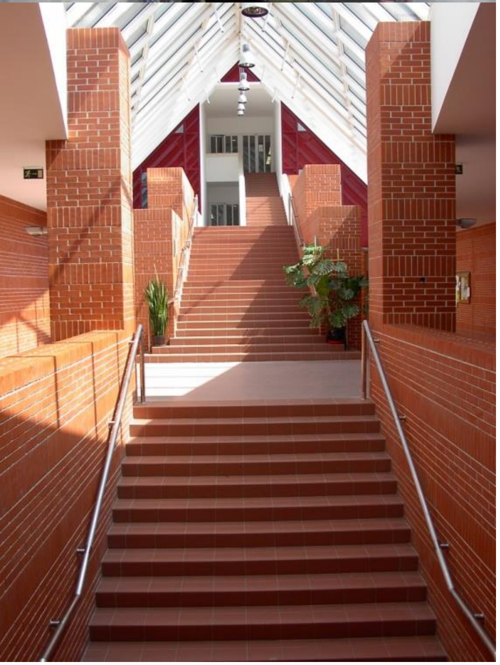
2. Keresünk **hátulról** egy elemet, aminek **elől** a helye (mert T tulajdonságú, nem odavaló):

   O x x x x x x **x** x x x x x

3. A megtalált elemet **tegyük** az előbb keletkezett **lyukba**:

   ⊗ x x x x x x O x x x x x

A lyuk mögött és az 1. elemmel már rendben vagyunk.



Szétválogatás helyben

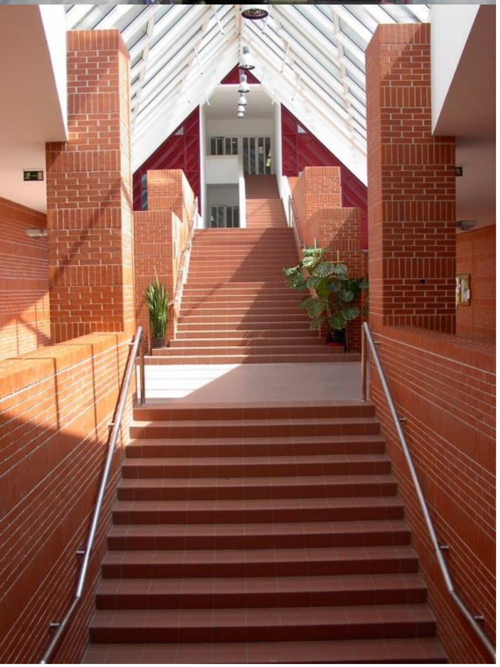
4. Most keletkezett egy lyuk hátul. Az előbb betöltött lyuktól indulva elölről keressünk hátra teendő (nem odavaló: nem T-tulajdonságú) elemet:

⊗ x x x x x O x x x x x

5. A megtalált elemet tegyük a hátul levő lyukba, majd újra hátulról kereshetünk!

⊗ x x O x x ⊗ x x x x x

Az elől keletkezett lyuk előttiek és a hátrébb mozgatott elemmel kezdve rendben vagyunk.



Szétválogatás helyben

6. … és így tovább …

7. Befejezzük a keresést, ha valahonnan elértük a lyukat. Erre a helyre a kivettet visszatesszük.

Szétválogatás helyben

Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
- Kimenet: $Db \in \mathbb{N}$, $X' \in H^N$
- Előfeltétel: –
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ és $X' \in \text{Permutáció}(X)$

  és $\forall i(1 \le i \le Db): T(X'_i)$

  és $\forall i(Db+1 \le i \le N): \text{nem } T(X'_i)$

Algoritmus:

Változó
e,u:**Egész**
y:TH
Van:**Logikai**

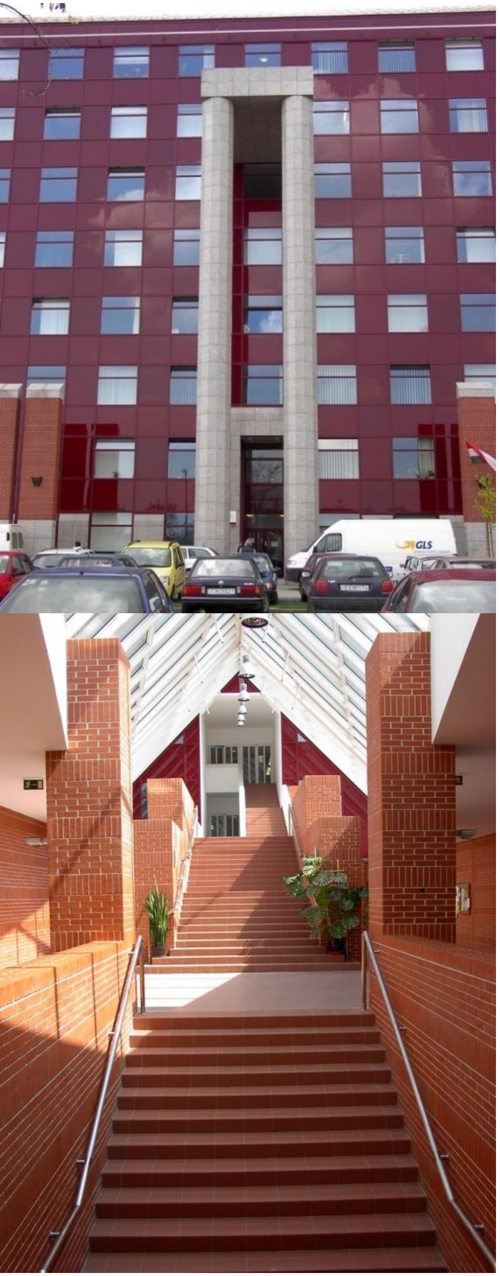
e:=1 [a szétválogatandók elsője]
u:=N [a szétválogatandók utolsója]
y:=X[e]
e<u
HátulrólKeres(e,u,Van)
Van — I:
X[e]:=X[u]
e:=e+1
ElölrőlKeres(e,u,Van)
Van — I:
X[u]:=X[e]
u:=u–1
Van — N: —
Van — N: —
...

Szétválogatás helyben

Specifikáció:

- Bemenet: $N \in \mathbb{N}$, $X \in H^N$
- Kimenet: $Db \in \mathbb{N}$, $X' \in H^N$
- Előfeltétel: –
- Utófeltétel: $Db = \sum_{\substack{i=1 \\ T(X_i)}}^{N} 1$ és $X' \in$ Permutáció(X)
  és $\forall i(1 \le i \le Db)$: $T(X'_i)$
  és $\forall i(Db+1 \le i \le N)$: nem $T(X'_i)$

...	
X[e]:=y	
I	T(y) N
Db:=e	Db:=e–1



Szétválogatás helyben

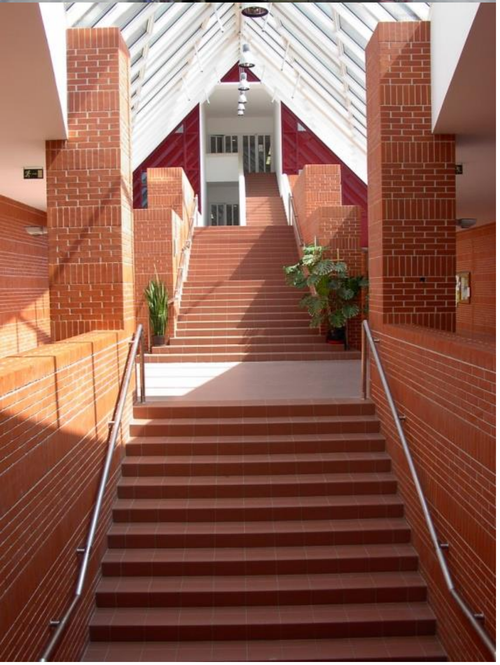
ElölrőlKeres(e,u:Egész,Van:Logikai)

e<u és T(X[e])	
	e:=e+1
Van:=e<u	

HátulrólKeres(e,u:Egész,Van:Logikai)

e<u és nem T(X[u])	
	u:=u–1
Van:=e<u	

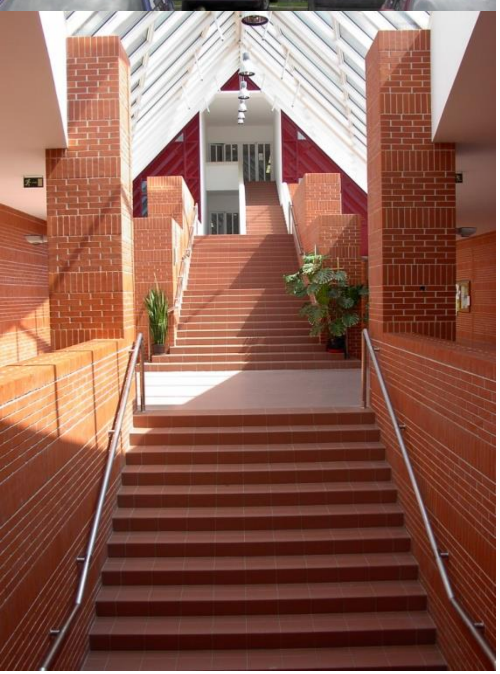
Hatékonyságvizsgálat táblázatkezelővel



Ötlet:

1. A táblázatkezelők importálnak sokféle formátumú fájlt, pl. CSV-formátumút.
2. A Comma Separated Value (CSV) = egy „mezei” text fájl, amelyben minden önálló (cellában tárolt/tárolandó) adatot (pontos)vessző követ.
3. Egyszerű olyan C++ programot írni, amely a táblázatolandó adatokat „CSV-esítve” ír text fájlba.

Hatékonyságvizsgálat táblázatkezelővel



Példafeladat:

Az unió és az összefuttatás tételek haté-konyságának összevetése.

Hatékonysági „dimenzió”:

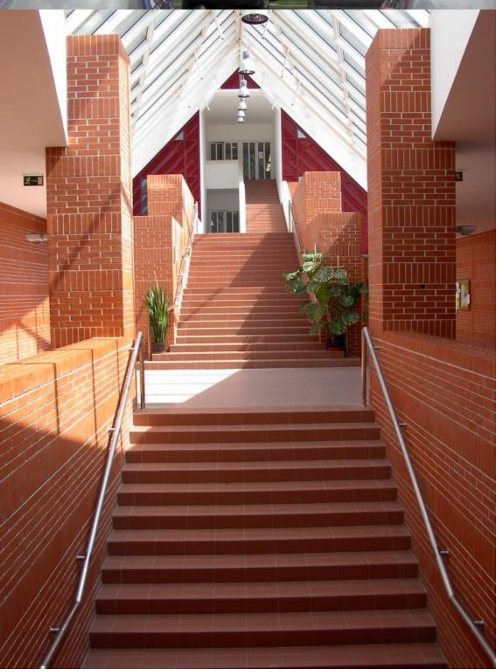
tömbbeli elemek összehasonlítás-száma

esetleg futási ideje (mint a futási jellemzője)

Összefüggést keresünk a bemeneti soro-zathossz és hasonlítás-szám között:

$(N,M) \rightarrow hDb_{unió}$, $(N,M) \rightarrow hDb_{összefuttatás}$

…

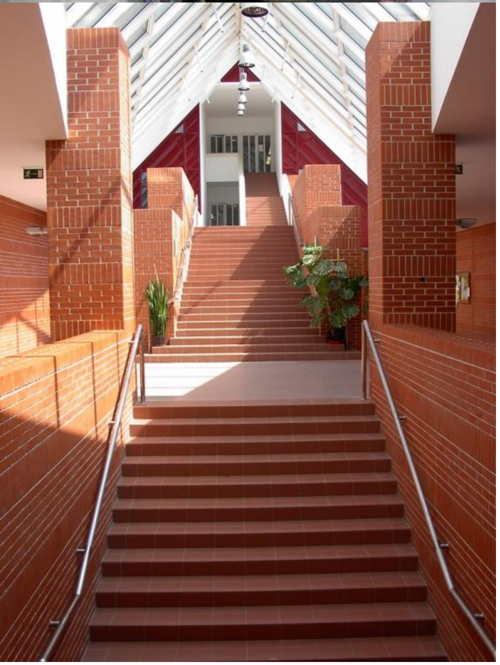
Hatékonyságvizsgálat táblázatkezelővel

Megoldásvázlat:

1. Mindkét algoritmusban számoljuk a tömb-elem-összehasonlításokat (mérjük az időt).
2. Néhány (jól kiválasztott) N,M-elemű sorozatra lefuttatjuk és közben számlálunk (mérünk).
3. Majd CSV-fájlba írjuk a hatékonysági eredményeinket.

Megjegyzés: az időmérés feltétele, hogy **pontosan** tudjuk mérni. (Windows-ban aggályos, Unix/Linuxban OK.)

Hatékonyságvizsgálat táblázatkezelővel

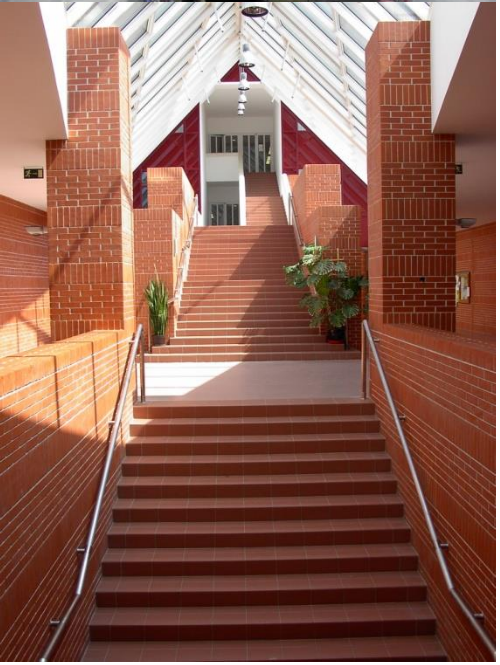


Egy lehetséges eredmény a táblázat-kezelőbe importálás után – unió:

Numerikusan

N \ M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0																				
1	0	1	2																		
2	0	2	3	5	7																
3	0	2	4	7	10	13	16														
4	0	3	5	9	12	16	19	23	27												
5	0	3	6	10	14	18	23	28	32	37	42										
6	0	4	7	12	16	**21**	26	31	37	43	48	54	60								
7	0	4	8	13	18	23	29	35	41	48	54	61	67	74	81						
8	0	5	9	15	20	26	32	39	45	52	60	67	74	82	90	97	105				
9	0	5	10	16	22	28	35	42	49	57	65	73	81	90	98	107	115	124	132		
10	0	6	11	18	24	31	38	46	53	62	70	79	88	97	106	115	125	134	144	153	163

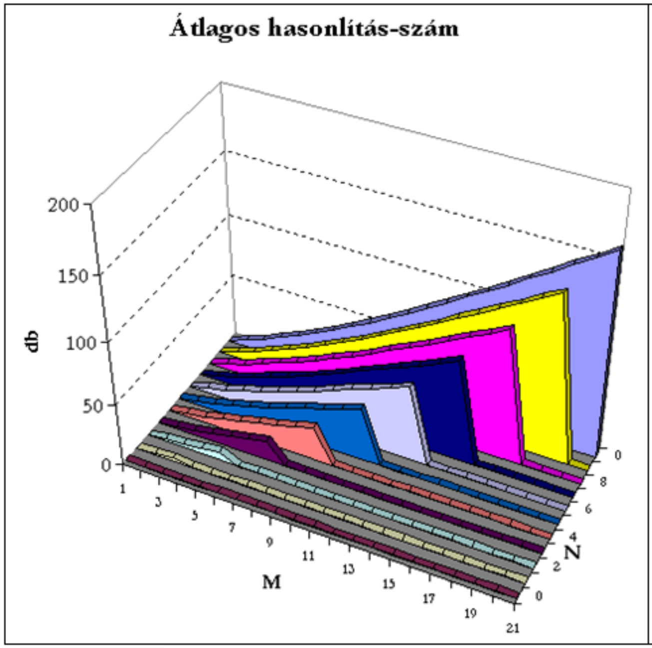
A program outputja táblázatkezelőbe importálás és némi szépítés után.



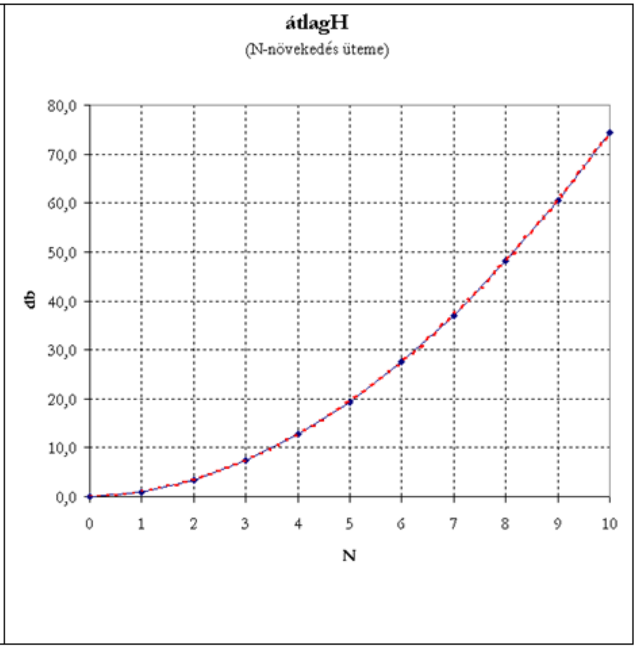
Hatékonyságvizsgálat táblázatkezelővel

Egy lehetséges eredmény a táblázat-kezelőbe importálás után – unió:

Grafikusan

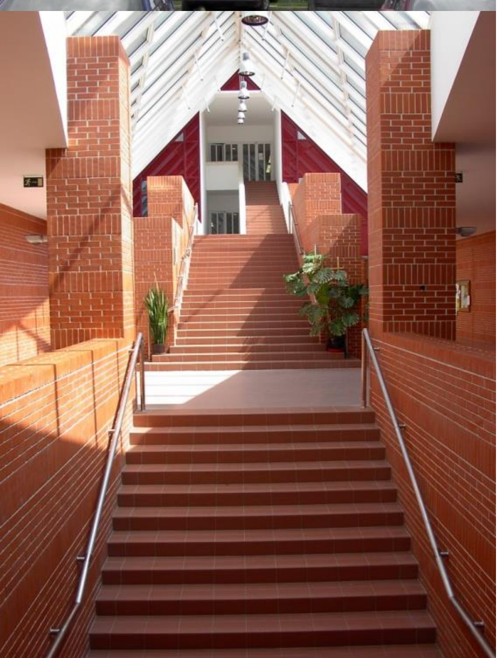


A program outputjának ábrázolása összetett felület-diagramokkal.



A program outputja és abból (rögzített N-hez tartozó) soronként képzett átlagok diagramja.

Hatékonyságvizsgálat táblázatkezelővel

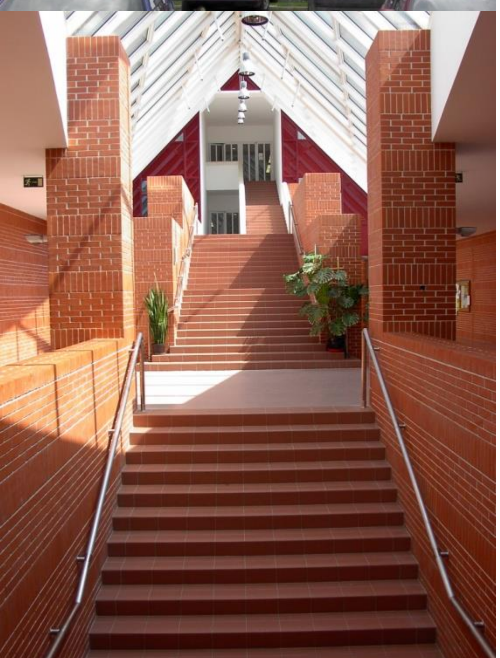


Egy lehetséges eredmény a táblázatkezelőbe importálás után – **összefuttatás**:

Numerikusan

N \ M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0																				
1	0	2	3																		
2	0	2	3	5	6																
3	0	3	4	5	6	8	9														
4	0	3	5	6	7	8	9	11	12												
5	0	4	5	7	8	9	10	11	12	14	15										
6	0	4	6	8	9	**10**	11	12	13	14	15	17	18								
7	0	5	7	8	10	11	12	13	14	15	16	17	18	20	21						
8	0	5	7	9	10	12	13	14	15	16	17	18	19	20	21	23	24				
9	0	6	8	10	11	13	14	15	16	17	18	19	20	21	22	23	24	26	27		
10	0	6	9	11	12	13	15	16	17	18	19	20	21	22	23	24	25	26	27	29	30

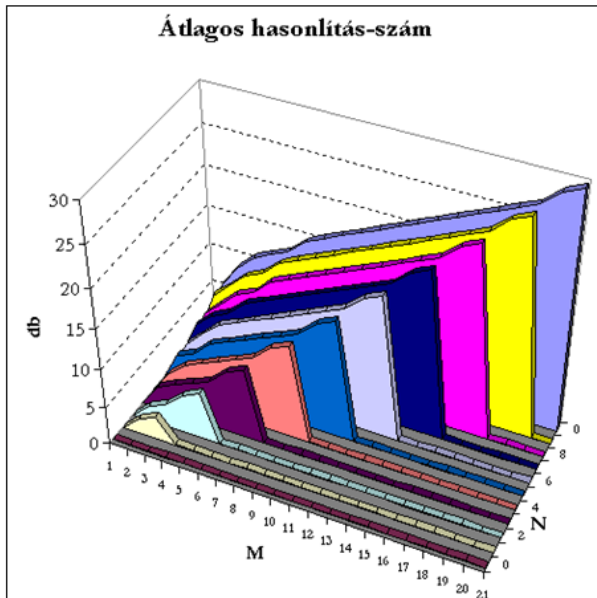
A program outputja táblázatkezelőbe importálás és némi szépítés után.



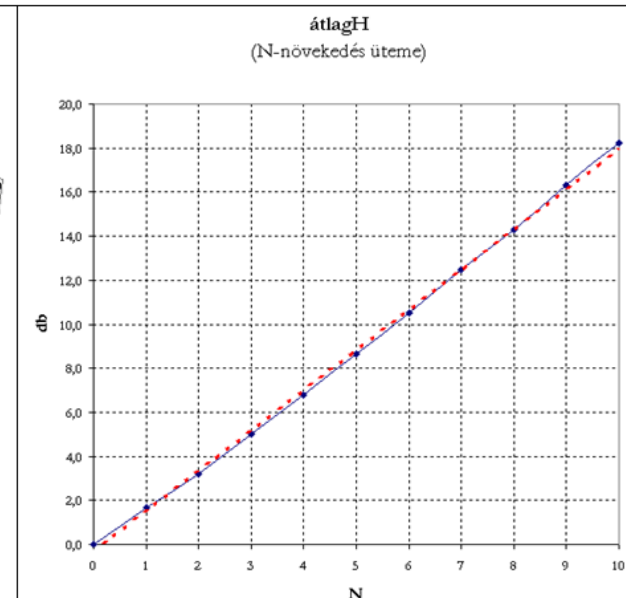
Hatékonyságvizsgálat táblázatkezelővel

Egy lehetséges eredmény
a táblázat-kezelőbe importálás után – **összefuttatás**:

Grafikusan



A program outputjának ábrázolása összetett felület-diagramokkal.



A program outputja és abból (rögzített N-hez tartozó) soronként képzett átlagolok diagramja. Az N növekedést mutató ábrán a legjobban ráilleszthető lineáris egyenes gráfja látható piros pontokkal.

Programozási alapismeretek
12. előadás vége