

aljo 11. EA.

ha sőt adat van, lassú a észlelés

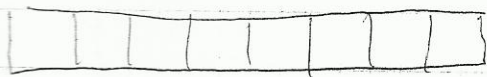
• a 3. típusú: a vége a végére kerül, de az  
E nem az eleje van, így mindig visszafelé  
léptetni, amennyi elem hiányzik

• h. típusú: megnevezhető az értelmezés:

mindig a sor, folytatja az eleje  $\Rightarrow$  az eleje  
magasabb címen van, mint a vége

vevőknél akkor fog, ha két van, vagy az  
E utoljára a  $W$ -t.

Son negatiivis



A son operatiivisen laulun selvennys, ohjelmien  
kell juttuilla kirkon... on elin...  
Ene laulujen... ohjelmien...  
kellon v. lista

Hoost kello.

Ole... kello... kello...

Konk... Max Elm = elin... kello...

Tilus Elm... kello... kello...

Son Til... kello

Elm: to... [1... kello...]

Elm, kello: Elm;

Kello

ELJÄRÄS SORTKED (Son-Sortin)

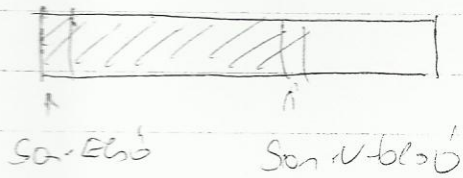
Son Elm := 1

Son kello := 0;

Kello

A son kello... kello... kello...  
kello... kello... kello...  
kello... kello... kello...  
kello...



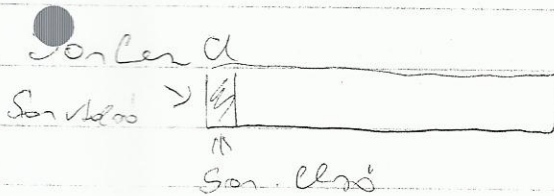


Elem felvétel

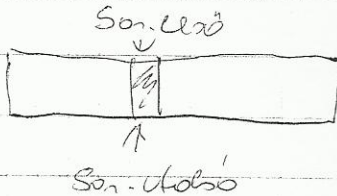
Sor. Utolsó  $\rightarrow$  Sor. Utolsó  $- 1$

Sor. Első [Sor. Utolsó]  $\rightarrow$  Adat

Elem kivételre FORDITVA

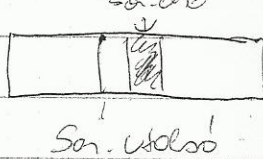


Ha elem csak elmozdítva el kell mozdítani



mind a két utolsó ugyanahhoz az elemhez

Definíció is igaz az  $\log_2$  első  $\rightarrow$  utolsó = a sor. Utolsó



FÜGGVÉNY Sor. Utolsó (Sor. Utolsó, Sor. Első) = log. Utolsó

Sor. Utolsó := (Sor. Első > Sor. Utolsó);

Frige


{ Elem csak az adott elemre } Sor

Ha a sor lényegesen olyan. Sor. első = helytől kezdve  
 feljebb  
 legkisebbért  
 érvényes  
 Sor. utolsó = legkisebbért  
 utoljára

Függvény Sorteli (Sor: Sortin): logika

Sorteli := (Sor. Utolsó = Max Elem);

Függvény

Er a 2 elemű sor  A két  
 sor, az első két részes és a 30 s  
 MA'SIK KÉT SOR

FÜGGVÉNY SORBÁ (Sor: Sortin, Adat: Elemi Valós)

Ha a sor Sorteli (Sor) akkor

Sor. Utolsó := Sor. Utolsó + 1;

Sor. Elem [Sor. Utolsó] := Adat;

Sora := 1 qm;

Kövéken

Sora := Keres;

Ha így "

Függvény

FÜGGVÉNY SORBÓL (Sor: Sortin, Adat: Elemi Valós)

Ha a sor Sorteli (Sor) akkor

Adat := Sor. Elem [Sor. Első]

Sor. Első := Sor. Első + 1

Sor. Első := 1 qm

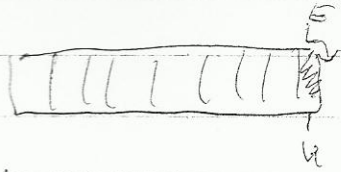
Kövéken  
 Sor. Első := Keres

Függvény  
 Függvény

Sor. Első := Keres



Egyes sorok a sor elején is lehet tele elemekkel



↓ Elemcsatlakozás

LEPTETŐ SOR:

FÜGGVÉNY LenSorok (Sor: SorTípus; Adat: ElemTípus):  
 logikai

Változó:  $l$ : egész;  $\{$  állás leírás  $\}$

Ha len Sorok (Sor) akkor

Adat := Sor.Elem [Sor.Eso]

Csoport := Sor.Eso in Sor.Utolso - 1

Sor.Eso := Sor.Elem [1+1]

csúcs

Sor.Utolso := Sor.Utolso - 1

LenSorok := lcs

előre

LenSorok := lcs

húzza

függ

LenSorok = (egyszerű) Sorok függvény

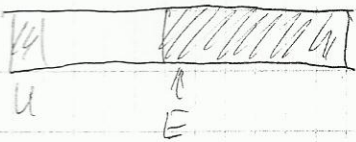
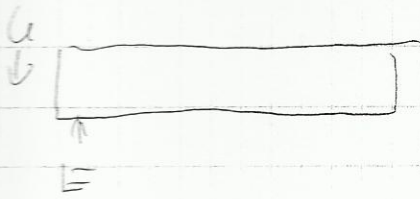
Ha az egyes elemek leírásai a gyűjtőbe  
 akkor a lista az utolsó a - elemet  
 az az első elem a Sor első pozíciója

Leírás a Sorok = egyszerű sorok

Ha az a lista a sorok és a lista az első  
 elem

# CIKLIKUS SOR

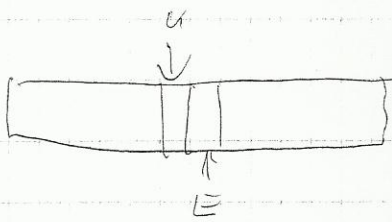
inicializálás után az elemek



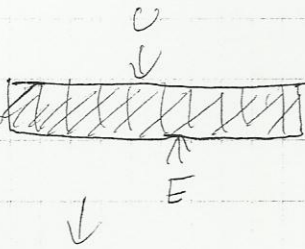
$E < u$  utolsó (IT MEMÓRIÁS)

Ha utolsó utolsó elemet a listán a végén  
és az utolsó elem

$$(u+1 = E)$$



ellenkező



$$u+1 = E$$

Ha az utolsó elemet a végén  
utolsó elemet a végén  
utolsó elem

Ha az utolsó elemet a végén  
utolsó elemet a végén  
utolsó elem

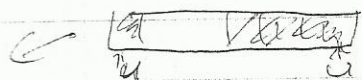
Értékes sor értéke az az az az utolsó elemet  
a végén az az az az utolsó elemet  
utolsó elemet a végén

FÜGGVÉNY Cse Sor Vés (sor: SorTIn): logikai  
CseSorVés := (Sor.Utolso = 0);  
függő



FÜGGVÉNY CÍLSORTÉLI (Sor: SorTíri): Logikai  
 CÍLSORTÉLI := (1 Sor: Első = 1 és Sor: Utolsó ~~akkor~~)  
 VAGY (Sor: Utolsó > 0 és Sor: Utolsó = Sor: Első - 1);

Függő



FÜGGVÉNY CÍLSOROK (Sor: SorTíri; Alet: ElemTíri): Logikai

Ha Nem CÍLSORTÉLI (Sor) Akkor

Ha Sor: Utolsó < MaxElem Akkor

Sor: Utolsó := Sor: Utolsó + 1

Különb

Sor: Utolsó := 1;

Műve

Sor: Elem [Sor: Utolsó] := Alet;

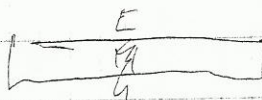
CÍLSOROK := 1; q := 1;

Írtör

CÍLSOROK := Kézi;

Műve

Függő



Függő CÍLSOROK (Sor: SorTíri; Alet: ElemTíri): Logikai

Ha Nem CÍLSOROK (Sor) Akkor

Alet := Sor: Elem [Sor: Első]

Ha Sor: Első = Sor: Utolsó Akkor

SorTíri (Sor);

Különb

Ha Sor: Első < MaxElem Akkor

Sor: Első := Sor: Első + 1

Kilometer

Son. Elb<sup>4</sup> := 1

Uvichy

Hude

~~atlas~~ Billenöl := (qu.)

Kilometer

Alkohol := (H. 10)

Hude

Frage

Son: Billeingutputten mit  
pilobiti bilid arleselun

el labon-L; markus

pi aylobon mit lyontyge boxy

Von ~~unhoripse~~

tegnitoc 2 fuyyungel listelöde  
uld uton fchis.

Lokis entordc farolix  
kityesi zuröbentich uny

Postfixes <sup>enteyes</sup> ~~avlet~~ listelöde

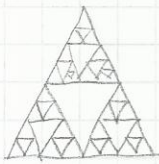
E = 111 11



Silpliuszi: önhasonló  $\Delta$ -er (van olyan része, amely hasonlít hozzá)

Adjunk meg  $P_1, P_2, P_3$  pontokat.  $\begin{pmatrix} P_2 \\ \Delta \\ P_1 \end{pmatrix}$

Kat. meg páronként az oldalfelezési pontokat.



lévő műveletet kellett végrehajtani, de sorozat  $\Rightarrow$   
REKURZÍV alakzatot kaptunk.

$\hookrightarrow$  ismétlődő jelent



$$\frac{3}{3} + \frac{1}{3} + \frac{4}{9} + \frac{16}{27}$$

$$u! := \begin{cases} 1 & \text{ha } u=0 \\ u \cdot (u-1)! & \text{ha } u > 0 \end{cases}$$

$$f_i := \begin{cases} i, & \text{ha } i \leq 1 \\ f_{i-1} + f_{i-2}, & \text{ha } i > 1 \end{cases}$$

$\hookrightarrow$  Fibonacci sorozat  $i$ -dik eleme

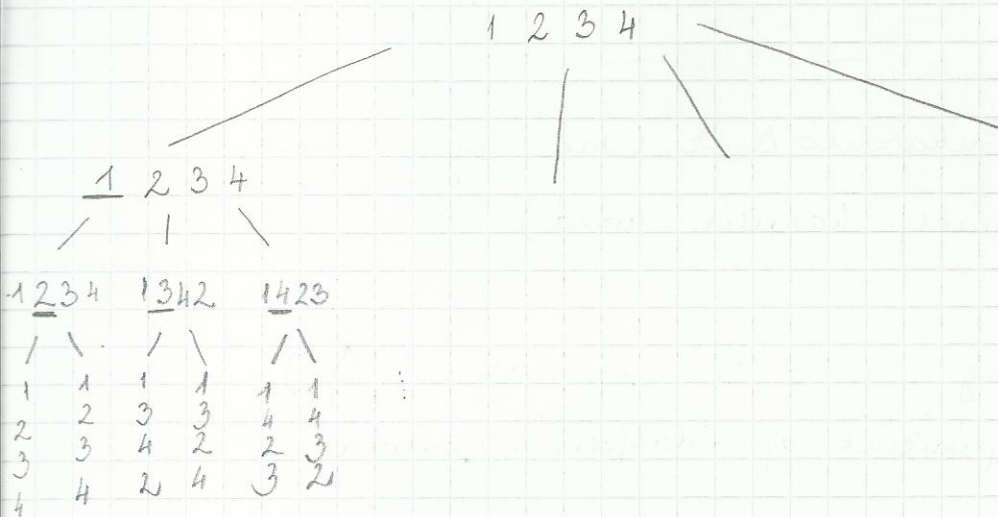
1 2 3 4     állítsuk elő a permutációját.

Mindenképp első rangjára az első helyen,

majd előállítsuk a maradék 3

permutációját.

Je



### REKURZÍV definíció

Olyan függvény-eljárást kell írni, amely saját magát hívja meg.

FÜGGVÉNY FAKT (N: EGÉSZ): EGÉSZ;

Ha  $N=0$  AKKOR FAKT = 1.

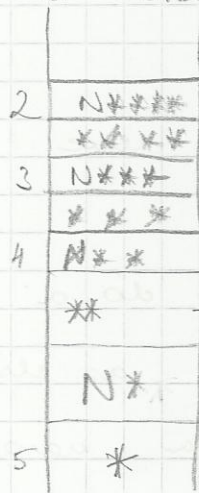
KÜLÖNBEN FAKT :=  $N * \text{FAKT}(N-1)$ ,

F VÉGE;

:

KI : FAK(5);

Nagy szerepe van a rekúrns:



→ etanolya, h. ide kell  
vissatérni, ha meg-  
hívásra kerül a  
FAKT függ.