

Alapfelte: - ki helytelenül hívják csapadék (pl. töltés esete van elcsúszás, illetve a térszámítás helyes töltésrendezés (q))

- we kann. reinerdeungesetzt, aber bei indunglich zu diet.

Pellid:
$$\begin{cases} A: = B+1 \\ C: = A+X \\ e: = B+1+X \end{cases} \quad (A^* \text{Hessinger volt})$$

- bizalmosabbi megismerés

- lóddido

Pellid: daltu - 12ptember - 4
 } listák helytől

leppüle - 3p. } aaduhel uuu hito
 2045 (indugitiditau) } nagy kutyák a helyet

Regiegele: Ft ten. bi hatdionpda aucter tódt. PASCAL eutq: ue-
 metitü a lauditide.

Hatellonpda

Globalis hatellonpda:

I. lbb - ciluok (az idb volterektok, a ciluok let vinqdus)

ciluo = leppidau * ciluonag futao ideje (a roppatbu balucluy
 cuuereuthes).

Lepponam volterete:

1. 000000 elemamnak volterete

(ha nem 12itadga, ne vinqdus, a 000000
 oroa elemet): chagou, amue uua illidg.

Pellid: nuuamoz: 2.. u-1 iq vinqdus, de edg 2.. u/2 iq
 vinqdus! de edg 2.. u_n, odt, 2, 3, .. in (paltan
 izamota).

2. 000000 kirete oitda:

(ogaitniku keredoi uddaj: uuaclg fcleris
 a vinqdusad elemel rodmot. (let reue oit-
 iu; letene euteg lotu reue is oitau).

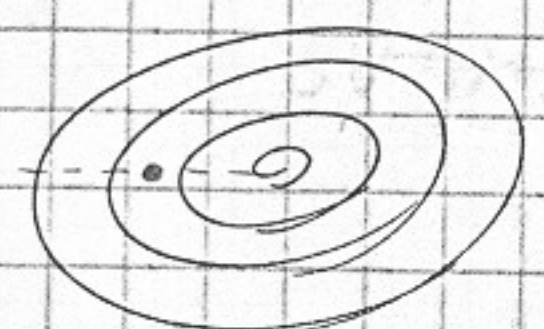
3. 000000k paituamoz kedolpoda: haladyuk eqyere a kedolpoda
 oonozatoban, ne vinqdus uua.

Gratouidg ierint: eideredo: ha a grauouibat hamarats megatallou,

a keratol idt at: aolka. Beadereie dt
 a2 eleuaret way, loay, a grauouibat
 honuz ebre. (Mit. neu ludij, est ebre.)

ejalio: ear ouardt kerendul a ait mekereatij, coellijit fur a2
 erite allioval. (neu auu. moqdot: eadueayemel).

Pelida: leuen earet a korepdt rira, a balhouan lephaua-
 labt edihetd rira.



Caporto: Holoqoza: Pelida: feruold oztalya, nere. Beaderes oztalya
 beuil heuoiba. equiare: n^2

oatayoument: $8 \cdot \left(\frac{n}{8}\right)^2 = \frac{n^2}{8}$

Pelida: vizalepeca kerend

Ciliau feruofomala: indereleoe:

Pelida: puzauuoz pelida: $I = 1$, $P = I$, puzauuoz *

Cikuro auuq $P \cap D \subseteq \bar{I}n$

$I = I + 1$
 $P = I$, $P \cap D \subseteq \bar{I}n$ *

lyeu fu. uiaa ! cikurod uiaamohato. Ataraluthato indereleoe.

$N \leq 2^{15}$

$N < 2^8$

$N < 200$

heylet egy puztombot adak weg: $P \neq P \cap P(I)$

Sokadq huz finouitaa

halmaz: $A = 2 * C$ (matizetael ha halmaz; halmazbaa az eleueluek
 uua oozidua).

olozat

Beuaris kuzaitas

2 eadueyt a beuarat hylet a mdt kuzaimoltdol aduallio. Pelida:

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$
 $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{k!(n-k)!} \cdot \frac{(k+1)!}{(k+1)!} = \frac{n!}{k!(n-k)!} \cdot \frac{(k+1)(k+2)\dots n}{(k+1)(k+2)\dots n}$
 Pelida: $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n!}{k!(n-k)!} \cdot \frac{(k+1)!}{(k+1)!} = \frac{n!}{k!(n-k)!} \cdot \frac{(k+1)(k+2)\dots n}{(k+1)(k+2)\dots n}$

els eaparetet $\cdot \frac{n-k}{k+1}$ (ue az alapadletthi uauuolka!)

5.1

PE

ILLUMAC FUTAI, IDEJE: Etágazás tervezésmódszer, tervezés

Pelda: 1000-1200 db-os, tervezés tervezés

II. HEL

1) Alap $I = 1-101$ $1000-1200$
Ha $V(I) = 1$ akkor $DR1 = DR1 + 1$
Ha $V(I) = 2$ akkor $DR2 = DR2 + 1$
:

cik. vég

ada

n_1 * helyett: $DR(V(I)) = DR(V(I)) + 1$ ada tervezés tervezés

tervezés tervezés tervezés tervezés tervezés

Adat

Pelda: Alap $I = 1-101$ $N-10$
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

HEM

Pelda: Alap $I = 1-101$ $N-10$
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

1. li

Pelda: Alap $I = 1-101$ $N-10$
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

cik. vég

PE

* helyett: $x(u+1) = T$ tervezés tervezés tervezés tervezés tervezés

tervezés tervezés tervezés tervezés tervezés

2. c

Pelda: Alap $I = 1-101$ $N-10$
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

cik. vég

PE

tervezés tervezés tervezés tervezés tervezés

Pelda: Alap $I = 1-101$ $N-10$
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

HEL

tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

3. H

tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor * tervezés tervezés tervezés tervezés tervezés
Ha $I = 1$ akkor tervezés tervezés tervezés tervezés tervezés

PE

* tervezés tervezés tervezés tervezés tervezés

Erőforrások megjelölése:

első példa

2	*	*	*
1	*	*	*
3	*	*	*

a 3-asok melle' számokot rajzolunk. A 3-asok helyett csak az index szerepel, és nem maga a 3-as.

GAJAS

programozás

(a feladat programjától eltérően)

helytelenül az új adatokból többet nem fogunk:

- elcsúszás
- elmaradás

számtani sorozat

$$u_n = \begin{cases} 1, & \text{ha } n = 0 \\ u_{n-1} + a, & \text{ha } n > 0 \end{cases}$$

$P(0) = 1$

$P(n) = 1 - 1/n$

$F(I) = I * I - 1$

$X = I * n$

alkalmazás

számtani sorozat

$R(I) = A(I) * A(I)$

$X = X + R(I)$

alkalmazás

alkalmazás, az idők eltelte után.

100 éves jubileum

$$P(x) = A_0 x^0 + A_1 x^1 + \dots + A_{n-1} x^{n-1} + A_n x^n$$

$$P(x) = 3x^{10000} + 2x^{1000} + 5$$

ne jöjjönnek fel, hanem
indulnak el, és az idő múlásával

(10000, 1), (1000, 2), (0, 5)

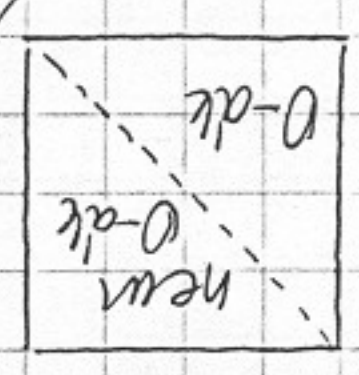
4. specialis operator

$x_1, x_2, x_3, \dots, x_n$ saját

eig tulajdonság: x_1, d_1, n

$d_1 \text{ leg } x_n = x_1 + (x-1)d$

Pelida:



$x_1, x_2, x_3, \dots, x_n$ atb.

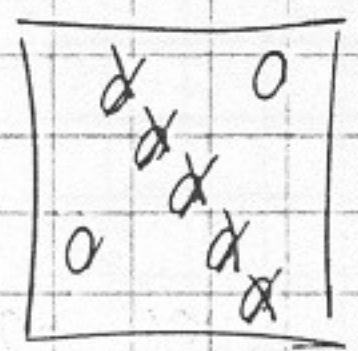
haazsodan az elso dhoz

carakoztasu higgoleges, ortogonalis.

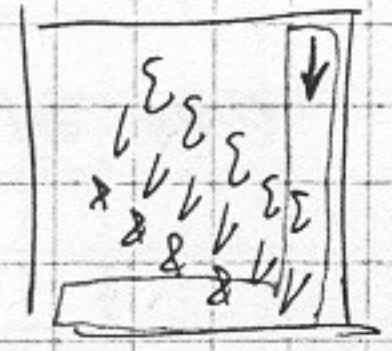
Nem matricot használunk, hanem helyettes egy matric.

n_1 rang linear. matrix: $X(x_i, y_j) = X(y_i, x_j)$ It eleg csak ezt a leso v. also idit tulaj.

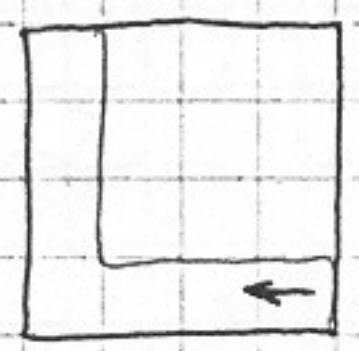
diagonalis matrix



topeltis matrix

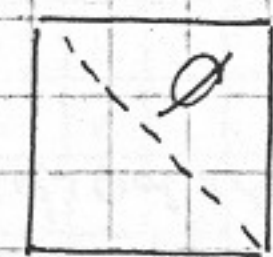


Hankel matrix



5. Adattérület megosztása

a leso egy matrixba drittelis vize.



Elemudef:

1. chagdas (adattérület egy részének eltagyasa, a tuit wakhonnyas megmaradnak.)
 Pelida: nem. idus + n. il. idis (teleleges)

2. osztas

amú malmolható, azt ne tholias teleleges.

Tarjola

Elida: luopu ueršet coraawae! teepuile; nevt iduuythiipawmal. (luopget heyyethenitietuik)

GRAMMISÖVEG

Elida

owoy heyyathilueatd iiepe. bouyueatag; caöiaeruteoe.

Adator eldheologagada

Elida: Paacabae a iidiöi, { atuagada, coeteg bilidaa eay builon hie-ke lgy a2 di adatorae uae quoraab a fedolqada.

Teleer öiseepitise

iiovegueret leaöiler, dit. hattekuoyatd

Ubd adate; traupformaliga

Elida: Program -> leatdhepauyo: Miteim ('...')

de hattekuoyabb eay eldie megit file hamaalata, elueuile a write kaap- adate. lgy tep i nevt kephte-keut, hauew iudolua ieruwie ueq, beu'staed.

NYOULTAG

outmaw adator

Algotimaw: i) Citikua bouyueatag; lya eay progawuabw eay citikua uaw, eay udeuileba; höök, adate a udeuilek bouyo- luttat:

elagandradu + ueruradu + 1

elrook: galto; citikua
houyueatag lya arowo
lgy a mat. detiucidruel).

idm nevt uideig ierueatö. Elida:

Ha f aluor ...

Ha f d; g aluor ... 2 (lax lue, 2 1/2)

Ha f aluor lya g aluor ... 3

szövegek összehasonlítása

1. szöveg = Referencia

DB: egység

Algoritmus: Lövsz (1. max DB, LPP)

Vege

Lap = Referencia

DB: egység

Algoritmus: Lövsz (1. max, fe)

rdge

fe = Lövsz (1. 2, pout)

Pout = Referencia

X1, Y1, Z: rdb

gyakorlatok példák leírása:

Tat = Referencia

DB: egység

Lap: Lövsz (1. max DB, leford)

DB: egység

Vonal: töv (1.)

Rozs, túl mélyen leírattam!

levegő

maximális méretű, a szöveg összesítője, mindenekelőtt meg lehet csinálni. minél több méretűt rak egy halmazba.

HATÉKONYSÁG

1. szöveg = Referencia

2. szöveg = Referencia

3. szöveg = Referencia

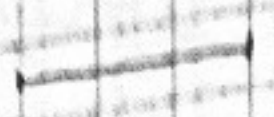
4. szöveg = Referencia

5. szöveg = Referencia

6. szöveg = Referencia

7. szöveg = Referencia

8. szöveg = Referencia



N

9. szöveg = Referencia

10. szöveg = Referencia

11. szöveg = Referencia

de- ea a mdrer? (iger - ege idt megal)

leqer az ebnet vllartot a 2. Ez allko, lenne id, ha mindet wogöte leaddone

Kvallogatás: $DB = 1$, $X(1) = 1$

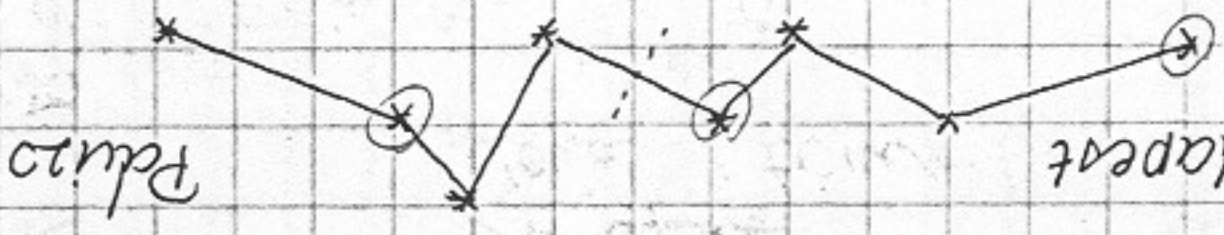
Ciklus $k = 2$ töl $N-g$ $X(DB)vege$ $Ha L(I) \geq V(X)vege$ akkor $DB = DB+1$, $X(DB) = 1$ cik. vege

eq. vege

leqacshb wdder leure: - öirer leshalma (2)
- kvallogatás
- maximumvllartotab

Altogat: egy adott idöpontig megtreim a müsozet kvalartaw. Az elöb it is allamართ leure (ara a reaktadatotla is).

Ida: Budapest



Imerük a beazinkutat helyet. Adyat meg a legkereset megdeshet.

Anol medllit az a öirer beazinkutat halma egy reahalmara, a leg- jök luyadonagq, legkereset elwagadant.

Kvalartowag = a let pont wööt edg a beaziy. Moho strategia: adag lee weur, a wug hucurk, eb igy tovabb. (Ha elbb hucokurk, akkor lera id, ha az elemzadur nem vllartit, wug esetben $N-g$ er wö).

Pelda: $X Ft!$

- 1000 Ft
- 500 Ft
- 100 Ft
- 100 Ft

kiezinkit az $X Ft$ ot minimumis ab dmever.

$Ha X > 1000 Ft \Rightarrow 1 ab 1000 Ft!$ $X = 1000$

$X < 1000 Ft \Rightarrow$

: haraldan a lötti duletel is

Altogat: leqer 500, 400, 300, 100 d Hieozink ki 700 Ft-ot:

Motto: 1 db 500 + 2 db 100; $\left\{ \begin{array}{l} \text{nem wudig id a MOHO wödirer.} \\ \text{de a megalda: } 400 + 300 \end{array} \right.$

Pelda

DI

jo m

leq

Pelda

Barabara poi alakbar dikir an arauqti, tujotot ...
 Nuyikibit menungit kee elurim, loay haring leqyen bebbie.

1 ab w_i (an i-edik aliyu) V_i (an i-edik elitke)

MAX (aruit a rabed elbir viumi)

1: V_i = $\frac{w_i}{w_i}$ = ruyequyi nuy elitke

2. An eldo elitk alapida readees noilkenub oareade. (w_1, w_2, \dots)
3. $w_1 + w_2 + \dots + w_i \leq \text{MAX}$, da $w_1 + w_2 + \dots + w_i + w_{i+1} > \text{MAX}$ (pakolius adaq aruy leket.

$$x = \text{MAX} - \sum_{j=1}^i w_j < w_{i+1}$$

amewup: warat meq, art a w_{i+1} -bi leqyik beke.

Et ar w_i . HATZAK Problema

tiak: $s = w(i)$; $i = 2$

--- Citra aruy $s \leq \text{MAX}$

$L_i: i-1, w(i-1)$

$s = s + w(i) \times A_2$ post a * leme.

$I: i = i + 1$

--- Cit. rige

$L_i: i-1, \text{MAX} - s + w(i-1)$

rige

da? leqyen an elabbi $w_i - 1$, an utolo $X + 1$ leqyen. elkos

Orj. elitk - $V_i + V_{i+1}$

eluwat utob wat

$V_i \geq \frac{1}{2} V_2 \geq \dots \geq \frac{1}{V_{i+1}}$ an edek wudane id, wimen id uat kaprak uat

ET Problema (HATZAK) = wun dabalhatu niut an elobbe.

(art neu id wudane; elwupida:)

$V_1 = 10, V_2 = 60$
 $V_2 = 20, V_3 = 100$
 $V_3 = 30, V_4 = 120$
 $\text{MAX} = 50$

$V_1 = 6, V_2 = 5, V_3 = 4$
 w_1, w_2, w_3

oid an elob leket viumi, leqyen a prios leq a id. A a wudobd neu id. $10 + 20 = 30 + 30 = 50$

Ha

otartóga: Pelda:



Kockákban forogó építők: Alkalmi a nagyságokat és a nehézségeket. (Ilyet hogy a nagyság, nem alakos a súlyoság).

Teadatok: Peldák használva a kockák és a súlyok a súlyok a nehézségeket.

vegyes: A törzsek használható mindegyikével a megfelelő súlyok (Kockák és súlyok).

A kockák nem is a nagyság, súlyok.

M_i (kockák száma) s_i (kockák súlya) $s_i \geq s_{i+1}$ rendezett sorrendben

és elte.

Pelda:

- ① 2 ③ ④ ... N

→ súlyok a nehézségeket

max. súlyok a nehézségeket

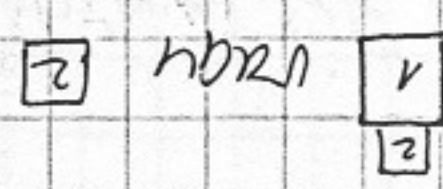
$Kocka(i)$ - a legnagyobb súlyú kockák sorozata, amelynek a legnagyobb kockája i -re illeszkedik.

Ha i -edik kocka van (melynek i -re illeszkedik), akkor az előző kockák a max. súlyok.

$Kocka(1) := M(1)$

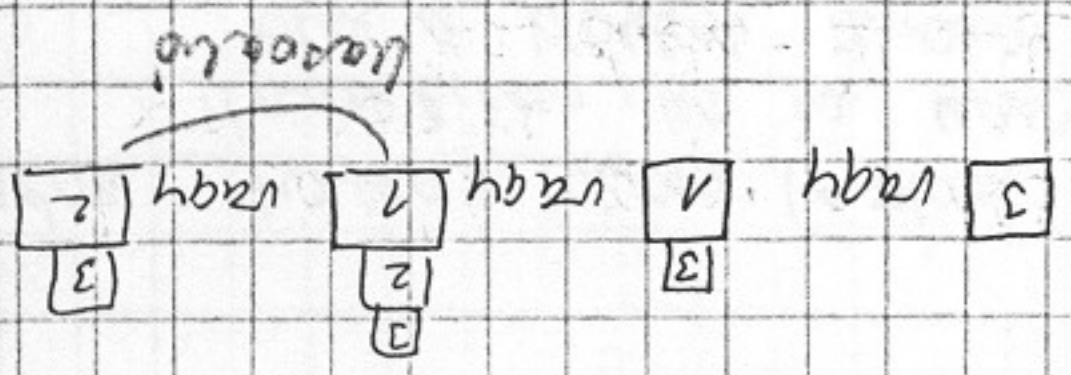


$Kocka(2) := \max \{ M(2), Kocka(1) + M(2) \}$



Ha $M(2) \leq M(1)$

$Kocka(3) := \max \{ M(3), Kocka(1) + M(3), Kocka(2) + M(3) \}$



(A kockák a legnagyobb súlyú kockák sorozata.)

kocka:

$Kocka(1) := M(1)$

Ciklus $i = 2$ - től N -ig

$Kocka(i) := M(i)$

Ciklus $q = 1$ - től $i-1$ -ig

Ha $M(i) \leq M(q)$

akkor $Kocka(i) < Kocka(q) + M(i)$

akkor $Kocka(i) := Kocka(q) + M(i)$

* $M(i) = q$

Lépcsők: $N - N - 1$

max. súlyok a nehézségeket

ely. súlyok

Ég a módszer a DINA Művészet programozás.

(melyek a kockák súlyok, melyek a nehézségeket)

(Héber: a nehézségeket a legnagyobb súlyú kockák sorozata)

Művészet

otartóga

otartóga

otartóga

otartóga

otartóga

otartóga

otartóga

otartóga

otartóga

otartóga

1. pr. az esb hia kell → nehezebb a megoldás. Helyig is táblázat. Hm, kihasználható all ell a többi elem. Ezt a sorozat a kell döntenem az egyéb.

táblázat matrix ⇒ 3 elem kell
 vector ⇒ 2 elem kell

adom meg, hogy a tolong, mita toltatásból épít fel? all adom minden bucskát, hogy mita temák rá * megkérdezek

do(s):
 Ha $MIRE(s) > 0$ akkor $MIRAS(s)$ relatív kúrd
 IR(s)

ingegbbi beállítások kiegészítések (az 1. állás 13. az.)

FILE-TÍPUS

Hírák híe
 logikai híe
 (adattípus igazodik) → (a hárdatör igazodik)

logikai híe-kat hírák híe kelle meg. önműködés → műveletek miatt hírák híe ide ide öndőben van használható.

file equige: hírák rekord (= blok): ami egy művelettel átvihető (praktikus, ha az nagy). Adattípus, letelező, letelező megkérdező) era a blokkok arányos hozzájárul. De a hírákban az új rekord az új rekord, az új rekord.

Híe equige: logikai rekord (≠ rekord adattípus-adj!) az. Hőreg-híe-adj 1 db rekord.

Hírák rekordok kőre logikai rekordok fel. Ditta az. 1-1 megkérdező rekord

tal blokku radmohatós, így kőrtteléni elhethet. Ezt az indexelési híe DIREKT azonosítást is veszti.

ek: Megnyit (F, ind), Azolva (F, I, P), Kír (F, I, P), LPAR (F)
 Hírák híe, a hírák híe, a hírák híe

An egyelő kiemelési műveletet a FILE LEÍRÓ REKORD-ÉRTÉKELÉS ALKALMAZÁSÁNAK EREDMÉNYEKÉNT ÉRTÉKELÉSÉRE VÉGEZIK. A FILE-LEÍRÓVAL ELŐZŐ ÉRTÉKELÉS MEGMÉRÉSÉRE VÉGEZIK. EREDMÉNYEKÉNT ÉRTÉKELÉSÉRE VÉGEZIK.

OGYAI FILE-OK

1. REKORDOK FILE: azonos sorban, egymás után elhelyezve. (Lásd a példaművelet, például a következőkben.)
 Képzés: i, j rekordok INPUT, i, j rekordok OUTPUT (Hálogadás: rekordok)

2. Direkt file: olyan file aminek leírásán az i -edik rekordja (Olyan sorban van, ahol a rekordok egy sorban vannak).
 (Hálogadás: rekordok, vagy direkt rekordok)

3. Átlósított file: azonos sorban vannak rekordok, de azonos rekordok nem mindig vannak egymás mellett.
 (Hálogadás: rekordok, vagy direkt rekordok)

A Húzó (input) file: a rekordok között van egy leírás a rekordok között.
 (Hálogadás: rekordok, vagy direkt rekordok)

REKORDOK FILE: a rekordok között van egy leírás a rekordok között.
 (Hálogadás: rekordok, vagy direkt rekordok)

OGYAI FILE-OK: a rekordok között van egy leírás a rekordok között.
 (Hálogadás: rekordok, vagy direkt rekordok)

OGYAI FILE-OK: a rekordok között van egy leírás a rekordok között.
 (Hálogadás: rekordok, vagy direkt rekordok)

ADATFELDOLGÓZÁS

(Most csak az a. kitérőre foglalkozunk)

1. MASÓLÁS: $f \rightarrow g$ (f felől \rightarrow g felé)

i) adatként (az utószó az adatként)
+ a beütéses alakok elnevezése (\Rightarrow kétértelmű, vagy kétséges)

ii) listázás (lista, leírás, leírás, leírás)
+ a beütéses alakok elnevezése

iii) struktúra váltás (másolás megváltoztatás egy felé)
pe: adatok elnevezése \rightarrow adatok elnevezése + új adat
 \rightarrow más oldal
 \rightarrow részlete pontos

iv) ábrázolás váltás
pe: másolás után új formátumra átalakítás

2. NY NY NY (NY a NY NY) : valamilyen részlet elnevezése
pe: másolás után új formátumra átalakítás
pl: másolás után új formátumra átalakítás

3. ALFALVÓT MASÓLÁS: leírás után új formátumra átalakítás
Lál liqáda:
pe: melyik oldalra van az adat \rightarrow átalakítás

4. ÖSSZEKÖZÉS: két eredményre egy felé

pe: pontos, összehasonlítás, egyenlőség, másolás, leírás

5. ÖSSZEKÖZÉS: egy felé az adatok megváltoztatás, amit követően kell
köls, beütés, és másolás

6. RENDEZÉS: két felé rendezés, az adatok megváltoztatás, amit követően kell
(NY NY NY) (NY NY NY) (NY NY NY)

F. VETŐLŐGÓZÁS

feladások: pe: felé \rightarrow ki: felé ($1-7$)

A feladatok megoldásai a következők: (NY NY NY) (NY NY NY) (NY NY NY)

Másolás feltevése az alapra az önmaguk:

Cikkü a magy. üvege? (z)
 Olv. (z, r)
 $z = f(x, r)$
 Cikkü üvege

Valóság meghatároz. a termék file-t, külön kiegészítőkkel.
 Ha $x = x_1, x_2$ akkor két fajta bontás.
 $z = f(x_1), f(x_2)$ azaz elemeként kiegészítés.

A termékét pr. lap nem lehet megcserélni.
 elemeként pr. lap nem lehet megcserélni.

Példa: Kivételként feltételek megadását.

Kivételként:
 $DB = 0; r = 1$
 $z = f(x, r)$
 Cikkü a magy. $x \in U$
 Cikkü a magy. $x \in U$
 $r = r + 1$
 Cikkü üvege
 Ha $r \in U$ akkor $DB = DB + 1, Y(DB) = X(r); r := r + 1$
 Cikkü üvege
 elemeként pr. lap nem lehet megcserélni.

Összeállítás: Kivételként itt meg kellene adni a bontás feltételeit.
 Összeállítás: Kivételként itt meg kellene adni a bontás feltételeit.

Strukturális megjelölés elve:
 ha elemek között van a bontás

elem: T halmazokból
 elem: T halmazokból
 elemek között van a bontás
 elemek között van a bontás

a be és ki elemek egy
 márt elemek között
 ha: az elemek között van a bontás
 elemek között van a bontás

Kiegészítés: a) Tagok: kiegészítés (kiegészítés van a bontás)

pr. magy. üvege kiegészítés: ha: elemek között van a bontás
 ki: kiegészítés (kiegészítés van a bontás)
 mi lesz megcserélni?

(Kiegészítés: elemek között van a bontás, ezért lehet megcserélni).

b) Örrekonstrukciós követelmény

Re: 2 adatot (örmeget kell tudni), de az adatokot levezethetjük

pl. löbt ültetbi alló laucok vaddoi iddieszhuu jümet de az ültetk

levezethet.

new lelet n-zer végigvesszi az adatokat - lacou!

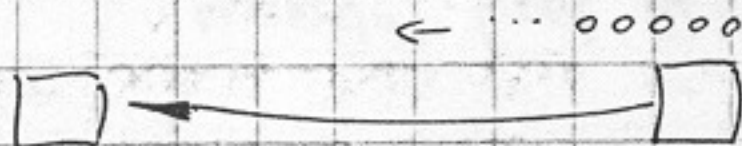
new lelet n fajr-ba egyezre lru -> opor. new enged.

bevezethet levezethet a file-ban.

-páratlan (egy iuduro vjara thoz. egy dlevo iddt talalva)

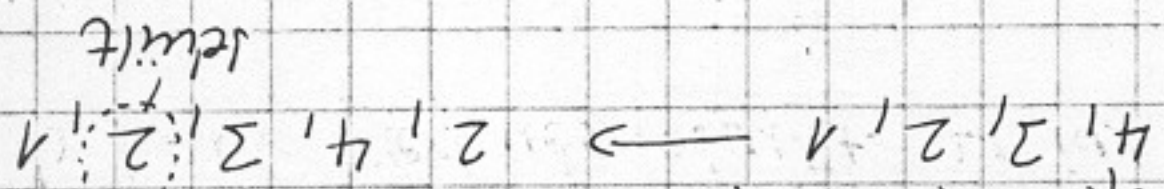
-leltre páratlan - páratlan halozat a datuvalertelmo, azo newagora

hottá.



raio, teko voral cetla ültetk a

adot newagot k.



stult

c) Árnyéklépcső követelmény: a bevezet és a visszavétel új elemek

most új elemek

- új elemek, az: növelés

Megoldás: fordított irányú a bit-t.

- csopotosan vanderit -> a csopotosat

úll casz vanderit és new elemek -

idut.

Tagoldási követelmények:

üi típus bevezetése: LAP, visszavétel: LAP, új

LAP, új

LAP, új

102 DB

L (n. max)

max. szám

hány új elem a típus

Lapozás

újt (f), lapozás (g) - new elemek idu

úllan új elem új elem (f)

úllan új elem (f, g)

LAP, új (g, g)

úll. új elem

úll (f), lapozás (g)

úll. új elem

Lapwyt (g)
Nyit (g)
LORD: = 0
Ei. vège

Laprat (g) sorz
Ha LORD = MAX akkor $L(g, L)$; $LORD = 1$
különb LORD = LORD + 1
 $L(LORD) = 0$

azt kell olyan hogy a lap után, csak ha a
beavult kiegészítés
(Göyelt: a beavult kiegészítés)

Laprat (g)
Ciklus a végé I = LORD + 1 - LBE MAX + g
 $L(I) = ' '$
Cik. vège
 $L(g, L)$
Ldr(g)
Ei. vège

(Hozható a blokkból olvasási, ideo)

Tagolási = Frikai - Logikai kité

(bokr \approx lap)

lapud: fejec, labec, lapozozar

0. elem MAX + elem + veltorod

örzetonddai konfliktus megoldása:

Palocitas: $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}_1, \bar{y}_2, \bar{y}_3$

$(x_1 y_1) (x_2 y_2) (x_3 y_3) (x_4 y_4)$

x_1, x_2, x_3, x_4 sor-stm vátokozat avok, labituet meg uicra pariza, ua
ior a pdr, liwewet a ereduwtytite ba

Palocitas eljeds \rightarrow lad asabb:

Palocitas:

Nyit (f, g)

Ciklus a végé new vège (f)

Olvas (f, r)

* Ha $R = X$ tpran, akkor aiba (e)

különbek sorból (a)

$k_i = (g, (a, e))$

Ha e párok nem lül wterre
vannak egyudatit, eger, li.
sor is eger.

Cik. vège
Ei. vège

Problema: $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$

Itäkiivä a 001; mitä vanha y is joutet eläkö?

* Ha uita a 001, allon, soiba (R)

T = R tipura

kuilöueta ha T = R hura allon, tonza (R)
kuilöueta tonza (R)

$r(q_1, q_2)$

Eqymas wöge ias: paioitaohoi mequualuato

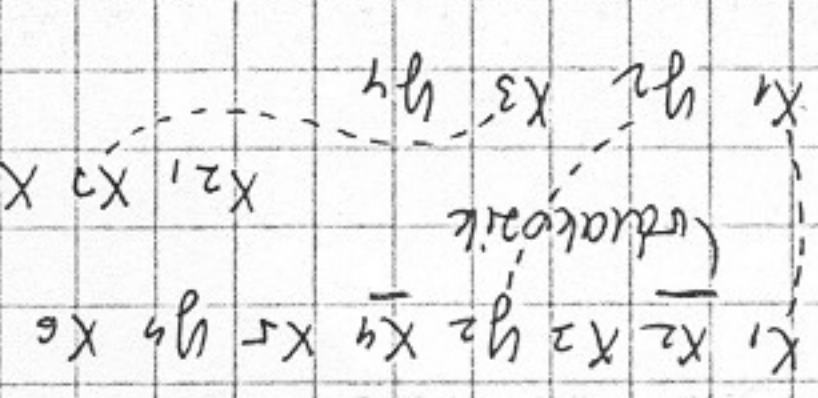
$x_1, x_2, x_3, \dots, x_n, y_1, y_2, \dots, y_n$

- direkt qyiltow eay soiba, wöge a eqet ort wäolow

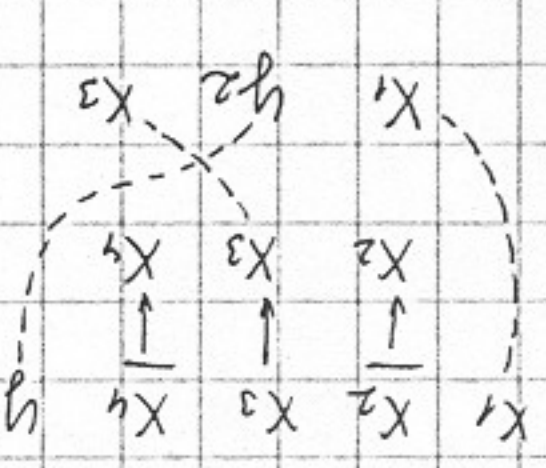
relat loay a 001 uay \rightarrow eläpöletti loay uay fäi e a
weuäiaban, allon ölay aoi leei aay a lätitit-

ou uay \rightarrow waceta)

Felteteta paioitaohoi:



Ha a soiba valaawoite id, allon a kiöu a soibde
foibol jöu ui: a, meqöu a päya
foibol jöu ui: a, meqöu a päya
↑
Ahol jöueta ui, allon a eläpöite eläp
vawuak



Felteteta paioitaohoi

uayt (f, g)
Cikuo auyg uay uay (f)

Oiva (f, R)
Ha a soi uita eö R id, allon ui (q, R)

kuilöueta Ha R uay, allon soiba (R)

kuilöueta ha R uay joutde, allon soiba (R)

foibol (R)

Cikuo auyg R uay uita e, eläp (001) jö

soibde (R)

ui (q, R)

Cikuo uay
uayt (f, g)
Ei. uay

Omawoada \rightarrow 001e allawanda

(aetwäoqatäs neu uatöuay)

Sörögfiladefiníció

Sörögfilé → Sörögfilé (székelyi fűszer)

Sörögfilé hprazol

1. Karakter

(karakterek elnevezése, leírás, stb.)
(nem feltétlenül ártalmatlan) - fűszerek

16 bites

5 bites

memória

memória
algebra
aritmetika

2. Sörög

Adaptívitás a karakterek
székelyi karakterek (székelyi)

fix

pl. string [10]

(Pascal)

LOGO

(a vége jel jelölés)

? nosz - sörögfilé i. d. d. d.

Működés

Hossz - karakterek száma

a) Indexelés (karakterek száma) ~ Pascal

b) Előző, Előzők, Utolsó ... ~ LOGO

c) Réz - indexelés ~ BASIC
ball (sz, i), kop (sz, i)

3 file

algebra

Ha a karakterek változó hosszúak, akkor a sörögfilé

3. Sörögfilé

(n karakter)

input < output

Működés

→ összes adat sörögfilé

1 karakter = 1 karakter

+ : sor (sor elején sörögfilé)

↳ sor végén el kell menni, sor végén sörögfilé

A sörögfilé-ban a sor végén van sörögfilé → sörögfilé

(Gondoljunk rá, hogy az adatok sor végén sörögfilé, sörögfilé)

Sörögfilé definíció

↳ sor, sor, sor, sor

input, output: fix karakter / sor

székely: min hprazol sor / sor / sor

↳ sörögfilé definíció

↳ karakterek

↳ sor - sörögfilé (székelyi)

↳ sor - sörögfilé → sor végén sörögfilé

↳ sor - sörögfilé

Működés mindig párosítva van

Karakter - sor

Első karakter (KAR, SZÓ) (- első karakter his vége a szó)

Szavéjre (SZÓ, KAR) (- a szó vége nem szerepel a szó végén, a szó végén nem szerepel a szó)

sor - sor

Első szó (SZÓ, SZÓ) A Képi a szó kezdet
Szavéjre (SZÓ, SZÓ) Bekesi a szó végét

SOR := SZÓ → első szó tő kezdés, mögött kezdés az elindítás:

Megvalósítás

Első szó (SZÓ, SZÓ)
SZÓ := " ; K := első (SZÓ) ; SZÓ := első karakter (SZÓ)
külön amíg K ≠ ' ' és K ≠ szó vég

SZÓ := SZÓ + K
K := első (SZÓ)
SZÓ := első karakter (SZÓ)

C.v.
Ej. vége

minden új szó vége a szó vége!

ABC / W XY

ABC X X X BE
↓
4, SZÓ

szó vége szavak felismerésénél

Szavéjre (SZÓ, SZÓ)

SZÓ := SZÓ + ' ' + SZÓ

Ej. vége

Lap vége (LAP, SZÓ)

LAP := LAP + SZÓ + szó vége

Ej. vége

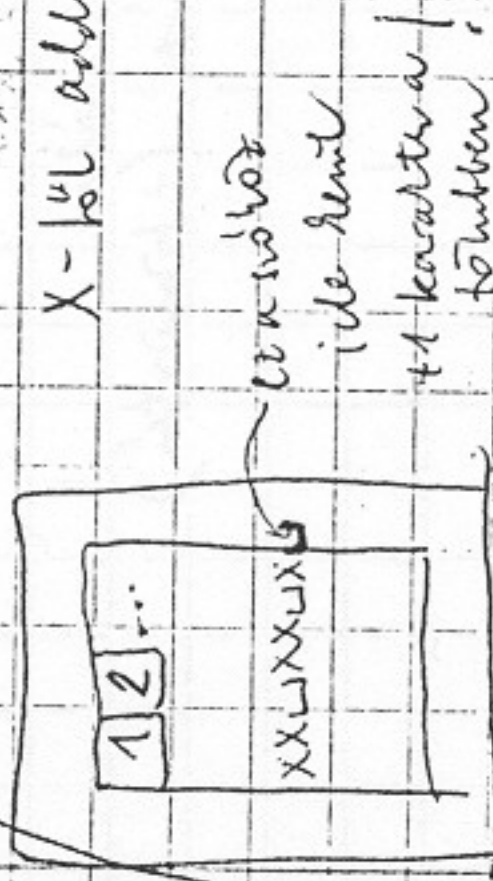
Képsor - mindig tartó képsor népszerű sor
történet (X-Y karakter) → oszlop száma (ha 1-181 indítom)

Képlap történet (A..B: képsor) → oszlop száma

Első képsor (képsor, SZÓ) → szóbeli kiértékelés 1. lépés

bejövő karaktert (belső karakter) és
el is kezdődött

X-vel adható, az új karakter



szó vége ide kerül + karakter a történet

(T)h

halmos kezdés szó, majd a képsor felismerése

Műt: A bejövő karakter népszerű

Képsor := " első szó (SZÓ, SZÓ)

külön amíg halmaz (KÉPSOR) + halmaz (SZÓ) ≤ KSHOSZ

KÉPSOR := KÉPSOR + SZÓ + ' '

első szó (SZÓ, SZÓ)

C.v.

SZÓ := SZÓ + ' ' + SZÓ (visztesztelés, ami nem feltétlenül első szó)

Ej. vége

Ha az a karakter, akkor bejövő lesz!

(Első szó nem mindig megkötés az első szó!)



A képsor vége általában más karakter

ha vége

szó vége szó vége

8.

Ha a mődönő's fajlban 4 rekord szer 1x szerepel,

ebből a 3. elemében a 2. helyen van a hibásolvasás

↓
Olvas (A, KA)
Olvas (B, RB)

1 elemet állt csak 1 darabot képezt!

(Alapötvejuttatásban nincs HIBA és hibák!)

Partner's wptdás: hibás mővellet kihi egy 4. fajlban.

HIBA: Kétféle ki (RB; D)

Ha B fél kőre fajl (pl 1 elemet elvett), akkor a felebi elemében

AA A fajl mővellet lemaradja, ez gondosajtklan

→ Gardaságs: pl havonta frissítésben

Alkalmazás: aronval kelene frissíteni → ez nem mővellet

Relevenciális fajlmal

↓
lásd Adatbővíítések

(itt nincs relevanciális fajlmal)

(Eveje az Adatfeldolgozás)

2H

→ Fajl, Adatfeldolgozás, hővefeldolgozás

↓
mővellet hi hely az alap relevanciális fajlban = pl visszafeltol olvas
blokkon blokkon

↓
Könnyűtőltő mővellet
lével példának

↓
Ez az a hibás mővellet
PASCAL mővelletben van indexelés

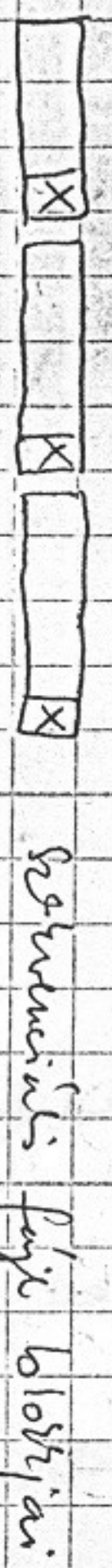
9.

LOGO-nem mővellet PASCAL nem alkalmaz

↓
Olyan ad: 1. növényi fajl konvertálásban

↓
Vgy mővellet mővellet kővellet → ad → bekérés → lap

KERESÉS



1) Rendelteti szemléltetési folyamat

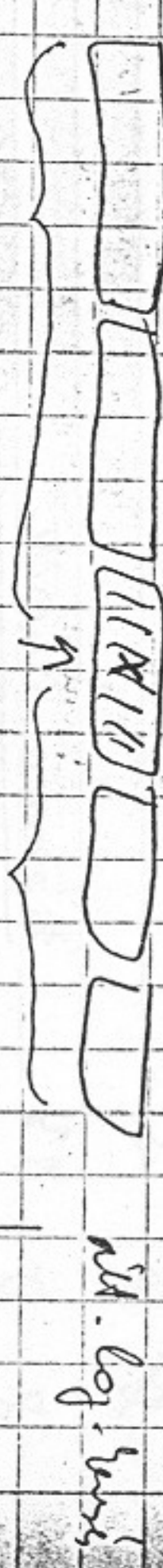
a) 1. blokkot ellenőrizni kell, nem volt az az érték, amelyet keressünk,

ha a szemléltetett elem nem az érték, akkor jár az

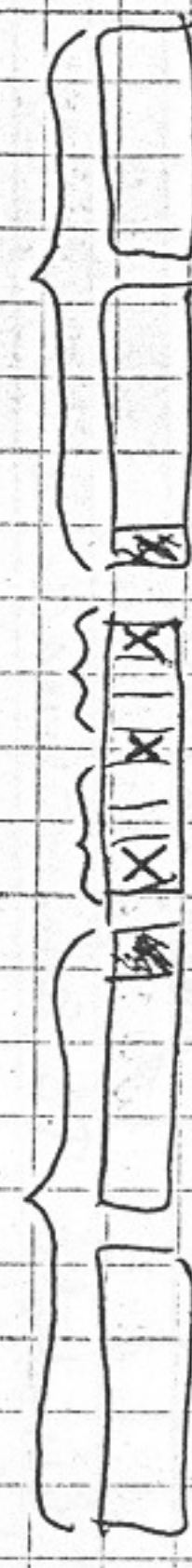
következő blokkra

b) blokkban belülről vizsgáljuk, mert lehet, hogy ott van az érték

2) Logaritmus keresés



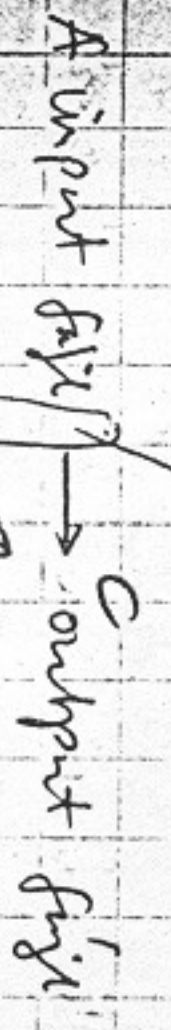
3. rész: logaritmus keresés 5. rész: eredeti sorrendű keresés



1. blokkot csak 1x kell beolvasni

Ezt csak akkor tudjuk meg, ha a szemléltetési folyamatot követjük

IDŐBÉLYKÉP



A input fájl

B módosított fájl

to get: V 3 fájl rendezett sorrendben "alapértelmezés" alapján

Időbélykép

Nyit (A, B, C): Olvas (A, B), Olvas (B, C) spec. a van a végén, azaz

Először RA ⊗ RB azin Ki (C, RA) : Olvas (A, RA) RA ⊕ RB azin Ki (A, RA) RA ⊗ RB azin Ki (B, RB) : Olvas (B, RB) : Olvas (A, RA) RA ⊗ RB azin Ki (B, RB) : Olvas (B, RB) : Olvas (A, RA)

Egy új RA ⊗ RB azin Ki (C, RA) : Olvas (A, RA) RA ⊕ RB azin Ki (A, RA) RA ⊗ RB azin Ki (B, RB) : Olvas (B, RB) : Olvas (A, RA)

Lu: Ki (C, RA) az a speciális: az azin az a végén

Egy új Ki (C, RA)

Ha hps (RB) = ujelenem, akkor Ki (C, RB)

Egy új hps (RB)

Ha hps (RB) = hps (A, RB) : Olvas (A, RA) Olvas (B, RB)

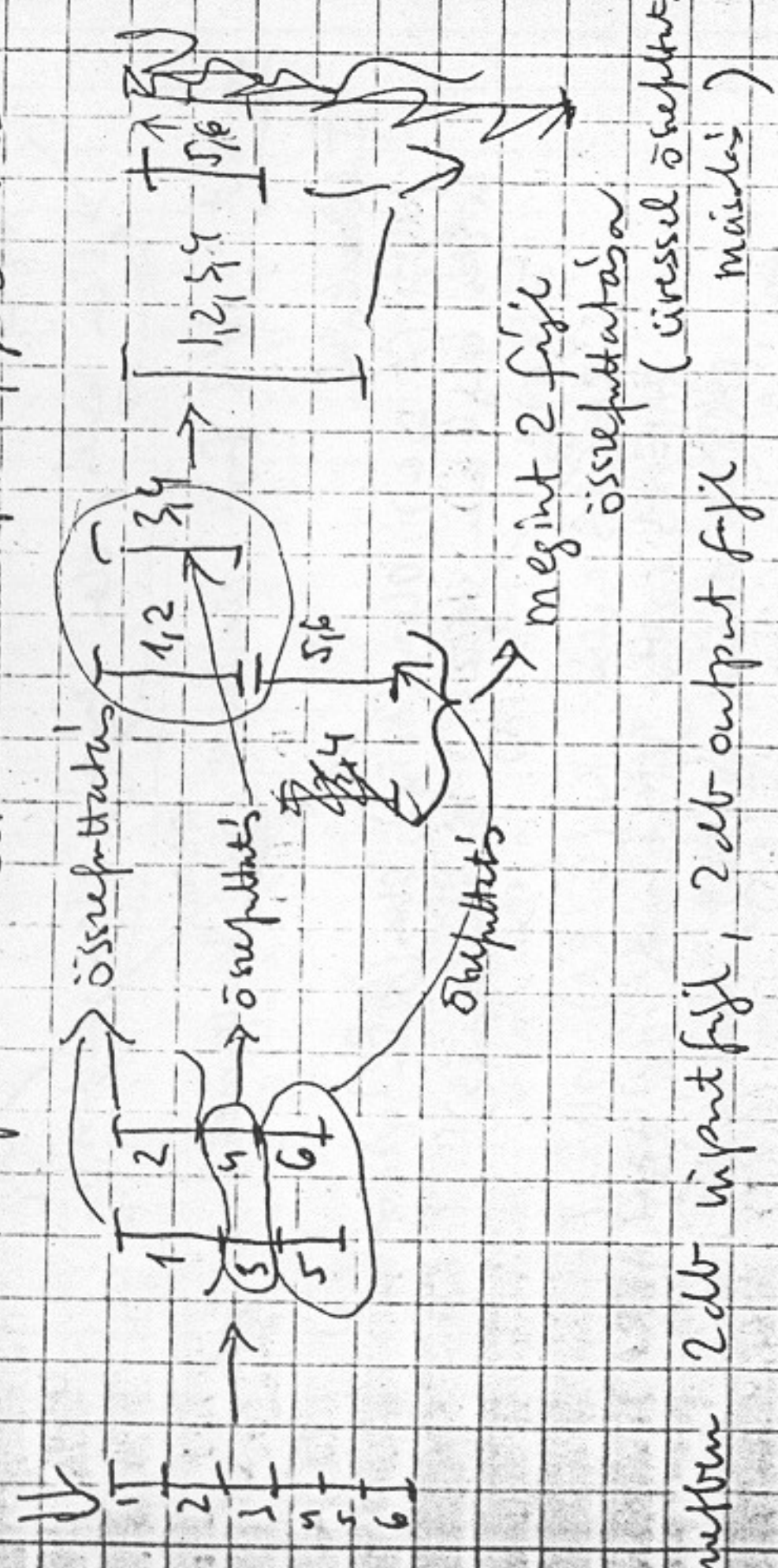
Ha hps (RB) = hps (B, RB) : Olvas (B, RB) : Olvas (A, RA)

Ha hps (RB) = hps (C, RB) : Olvas (C, RB) : Olvas (A, RA)

Ha hps (RB) = hps (A, B, C) : Olvas (A, RA) Olvas (B, RB) : Olvas (C, RB)

4. a) 4 segédfájlo tudas (→ helyre állított nem kell tudni)
 a) összerakás párosként
 b) összerakás szóval

→ op. No. korlátlan - milyen fajlól rakjuk.



megint 2 fájlt összerakásban (üressel összerakás = másolás)

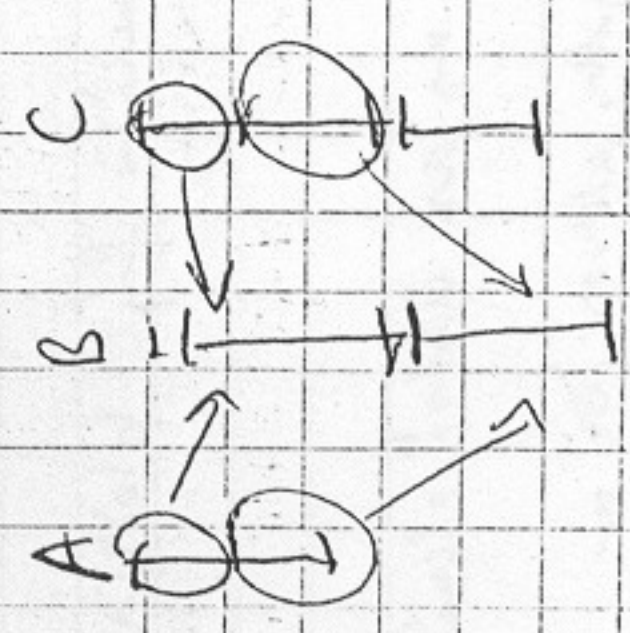
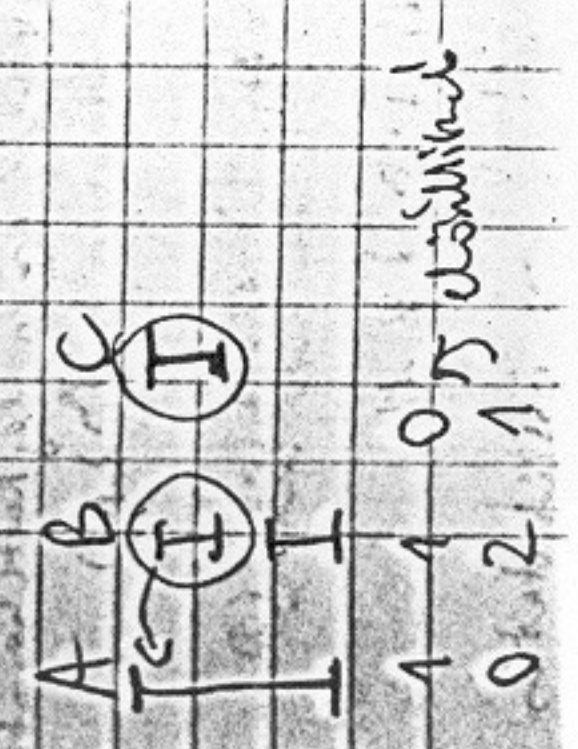
A mentem 2 db input fájlt, 2 db output fájlt
 legvégén: 1 db rendelt output

összerakás nem érétegy a fájli méretére, éppen csak 1-1 elemet kell tárolni

4.1 3 segédfájlos rendezés

A₁, B₁, C₁ A₁, B összerakás C-be

1 db 1 0 → összerakás előtti rendezett sorozat
 2 0 1 → összerakás utáni rendezett sorozat



a maradót a végén nem másoljuk át (csak a két)

2 0 3

5 3 0

8 8 5

8 8 13

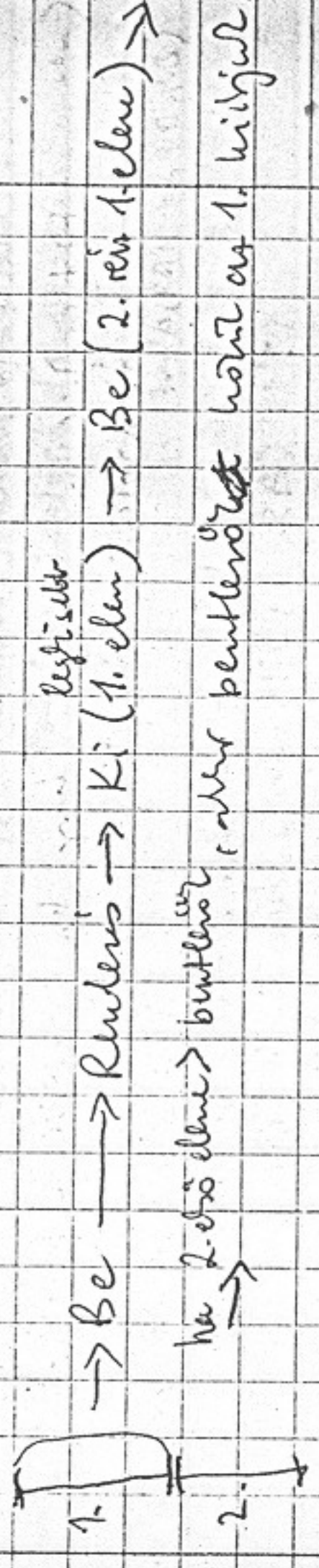
21 13 8

Fibonacci számok
 vannak 2 segédfájltan

amennyi van a darabokból

Ha a kívülről DB-szerű Fibonacci szám, akkor az alapján eléri a DB1-állapotot

Működésüket lehet rendezni, mint az a rendezésben elfut



Font: ha az új beáramlás kisebb, mint az első beáramlás, akkor új fájlt rendez

Felkettles párosítás

(T₁ a g₁ párosítás)

Nyit (f, g)

Ciklus és nem vgl (f)

divas (f, R)

Ha a sor az $\rightarrow R$ j₀, azu ki (g, R)

Ha R fozza, azu sorok (R)

Ha R nem j₀, azu

ki (g, R)

(k₀ x₀ t) \rightarrow Sorok (R)

*

* Ciklus and a sor nem az \rightarrow (g₁ sor) j₀

sortok (R)

ki (g, R)

ciklus

az u vgl

Zár (f, g)

f₀, g₀

Összeadás \rightarrow sorok adalása (számlálás nem kell)

legkevesebb \rightarrow mit

2005.04.20

3. Rendeltetéségi kódkódok

benne & benne sorok is van felnyitással

be \leftrightarrow ki (m₀ sorokból van kezdés)

1) Fordított sorok

Rendezés

Nyit (f) Nyit (g)

Ciklus and nem vgl (f)

divas (f, R); vége (R)

cur

1. c_i és a_j nem üres VEREM
 Vennböl (R), $K_i(g_i, R)$
 c.v.
 Zár (f, g)
 Fj. véje

Sajnos: adatak üresen kell lenni, azaz a fájl \Rightarrow
 \Rightarrow Verem hárterjárón is meg kell tudni választani
 \rightarrow hárterjáról visszatérés kell tudni olvasni, ez új fájl fájl!

(Nézzük 2N lépésben vissza)
 B) Szoportoként rendeztek
 pl. autógépjáró a jépek

1. Daraboként rendeztek
2. \rightarrow az 1. oldalra írjuk az első 2-ontól jobbra
 de nem ki

A darabok majd nem mindig elférnek a memóriában
 A felismerésük 1-pel előbbre olvasunk
 szoportoként olvasunk, szoportoként rendeztek,
 szoportoként írjuk az eredeti fájlban

B) Majdnem rendezett
 pl. sportaj fidelekénti eredetileg
 1. fordulóban max 2-t ugráz a listában
 (utolsóból nem lesz első utolsó)

2. Legfeljebb $\langle K \rangle$ helyre lehet átvinni
 (Kt1. min nem lehet első!)
 első K-1-t prioritási sorban rendezem

Rendezés
 Nyit (f) Nyit (g)
 Cikk I := 1-től K-ig
 Olvas (f, R); Prsorba (R)
 c.v.
 Cikk amíg nem vége (f)
 Prsorba (R); Ki (g, R)
 Olvas (f, R); Prsorba (R)
 c.v.
 Cikk I := 1-től K-ig
 Prsorba (R); Ki (g, R)
 c.v.
 Zár (f, g)
 Fj. véje

A prioritás; sor K elemeket - ez nem til meg

RENDEZÉS HÁTERJÁRÓN

Hárterjáró: az a hárterjáró, hogy hányra kell beolvasni a kártyákról
 (sorral lassabban, mint hányra írjuk a memóriába - pl
 hárterjáró)

\rightarrow ~ háttérjárási módra \rightarrow fájlra írjuk
 (blokkolvasás)
 blokkolás

Fontos: a) t elemet be kell olvasni N db olvasás } min 2 * N
 b) t elemet ki kell írni N db írás

Ha a fájl elfér a memóriában teljesen, akkor ez a legkisebb hárterjáró
 ($\approx 2N$)

Sorok: Nagy fájl, mi nem fér el a memóriában.

1.	2.	3.	4.
----	----	----	----

 \rightarrow lényegesen kisebb K db
 Hárterjáró fájlban \rightarrow összehasonlítás
 Rendszeri Rendszeri
 K₁ K₂

EF: van K db rekord
 3.

sor első elemét nem először
 meg kell írni