

ADATSZERKEZETEK ÉS ALGORITMUSOK

Külső rendező

Külső rendezések

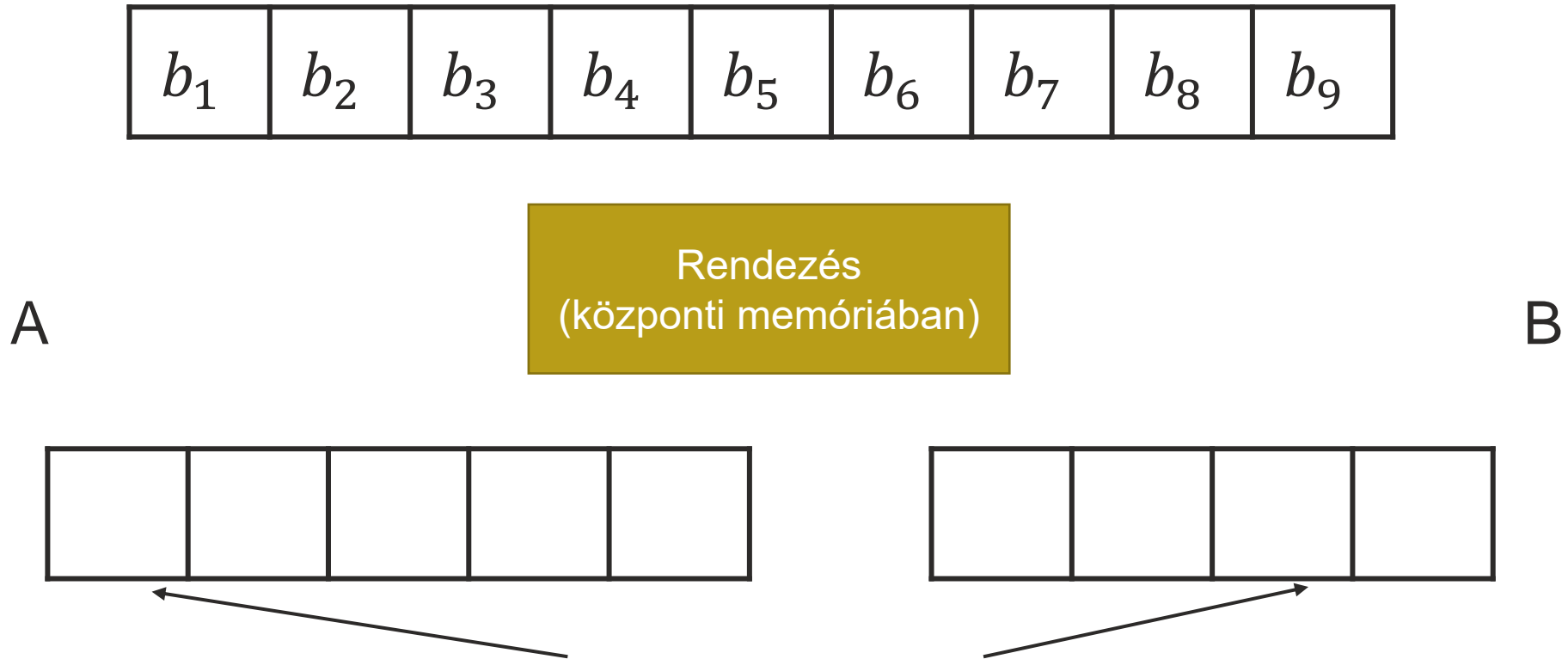
- Az összefésüléses rendezés külső táraikon
 - Adott egy S szekvenciális input fájl, amely n blokkból áll, minden blokkban adott számú rekorddal.
 - Pl. 1 blokk = 1024 byte és ezen 4 rekord foglal helyet.
 - A blokkok tartalma rendezetlen.
 - Az összefésülést iteratív módon végezzük, úgy, hogy az egyes „menetek” végén egyre nagyobb darabok, vagyis egyre több szomszédos blokk lesz rendezett.
 - Az összefésülést menetenként váltakozva az A , B , illetve a C, D fájllokba végezzük, végül a teljesen rendezett eredményt S -be írjuk. A közbülső menetekben azért lesz 2 output fájl, mert az összefuttatás eredményét az „egyet ide, egyet oda” elv alapján írjuk ki.

Külső rendezések

- 1. menet
 - Beolvassuk S rendezetlen blokkjait, valamilyen belső rendezővel rendezzük, majd kiírjuk felváltva A-ba, illetve B-be.
 - Itt még nem volt összefésülés.

Külső rendezések

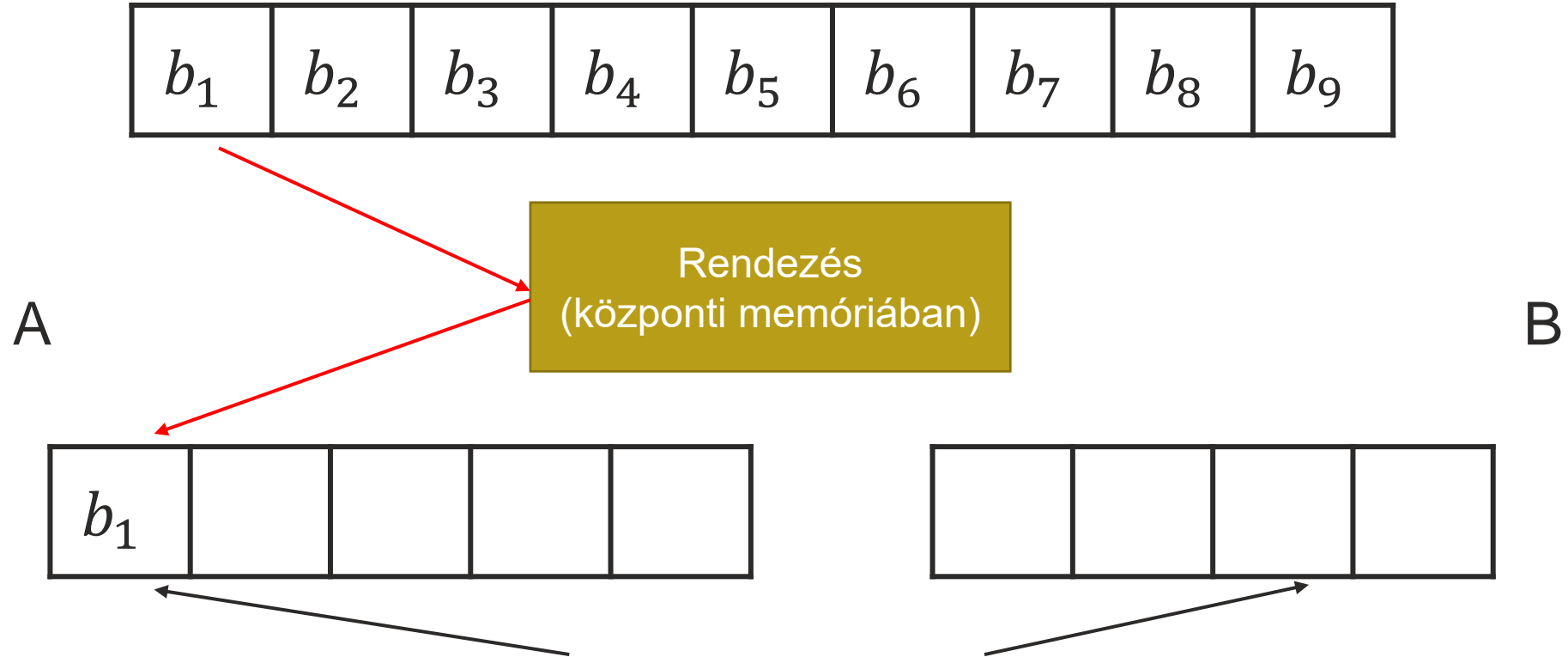
- S: rendezetlen blokkok



Önmagában rendezett blokkokat tartalmaznak

Külső rendezések

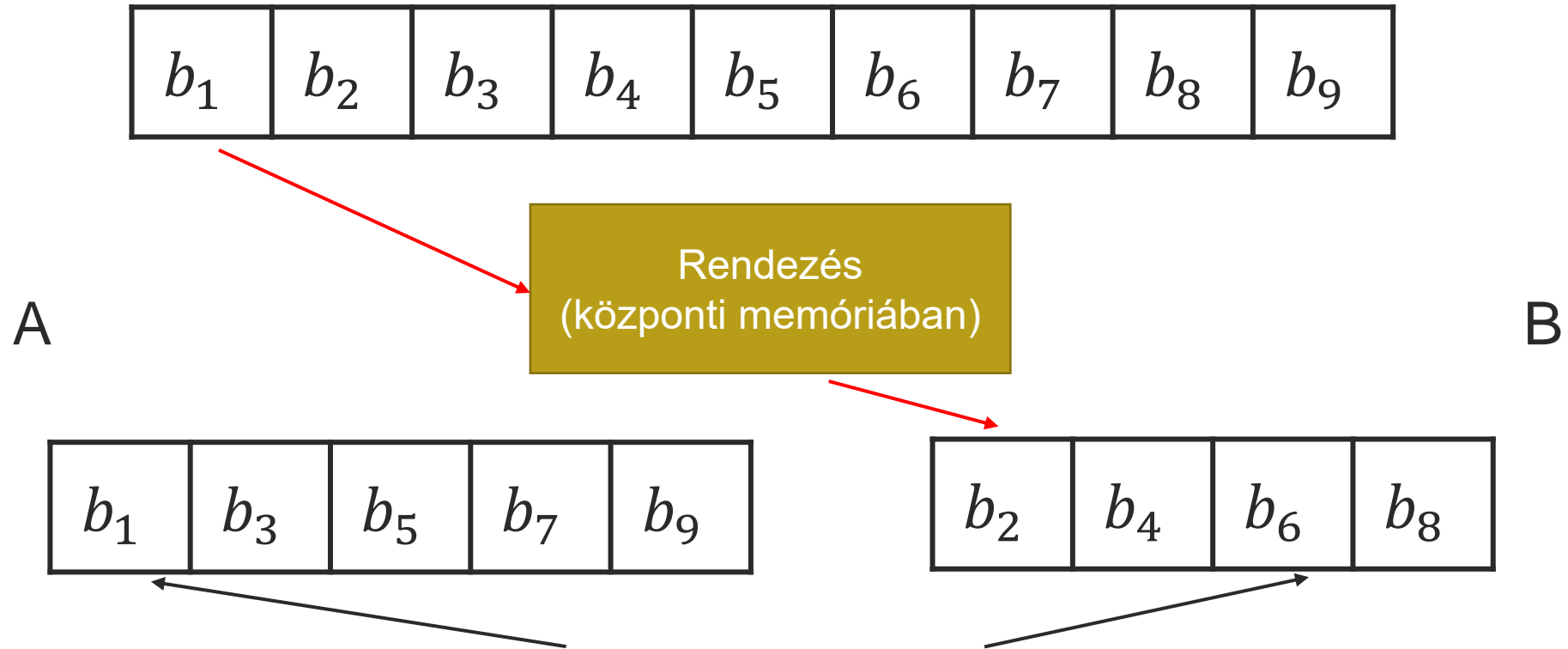
- S: rendezetlen blokkok



Önmagában rendezett blokkokat tartalmaznak

Külső rendezések

- S: rendezetlen blokkok



Önmagában rendezett blokkokat tartalmazznak

Külső rendezések

- 2. menet

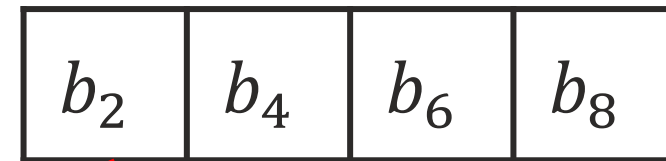
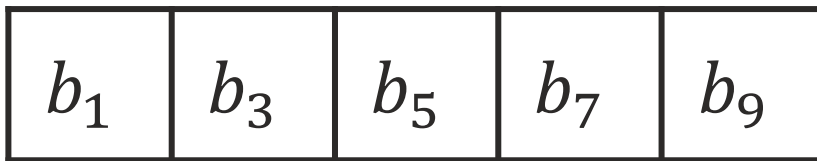
- Sorban beolvassuk A és B 1-1 blokkját. Ezek rendezettek. Összefésűljük őket és a rendezett két blokk hosszú adatot felváltva C-be, illetve D-be írjuk.
- Az A utolsó blokkjának nincs párja, így azt kiírjuk C-be. A C és D fájlban a rendezett részek hossza 2 blokk, illetve a maradék esetében 1 blokk.

Külső rendezések

A

Rendezett: 1 blokk

B



C

Összefuttatás
(RAM)

D



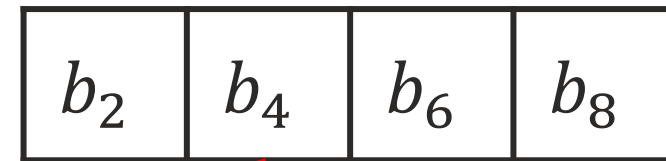
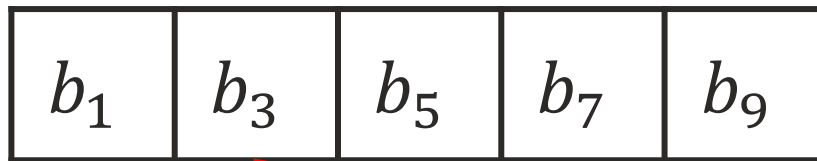
Rendezett: 2 blokk

Külső rendezések

A

Rendezett: 1 blokk

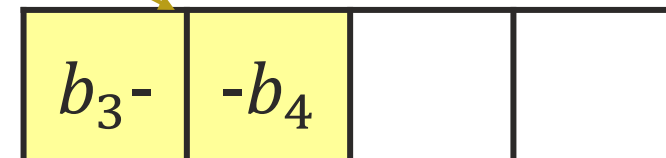
B



Összefuttatás
(RAM)

C

D



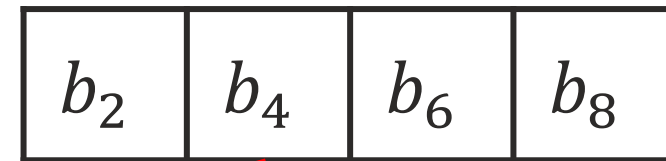
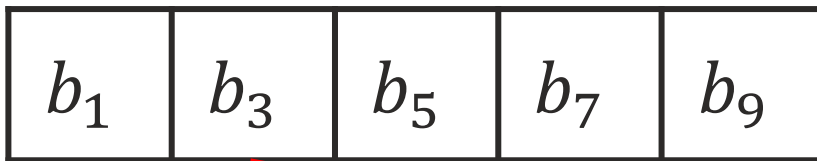
Rendezett: 2 blokk

Külső rendezések

A

Rendezett: 1 blokk

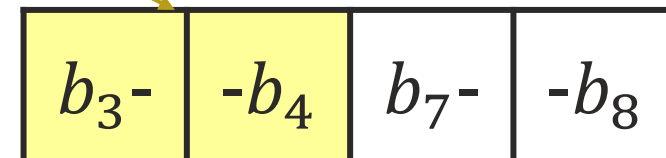
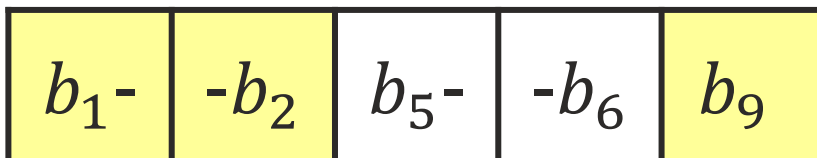
B



C

Összefuttatás
(RAM)

D



Rendezett: 2 blokk

Külső rendezések

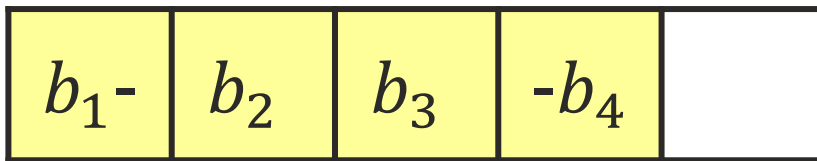
- 3. menet
 - C-ből és D-ből olvasunk 2-2 rendezett blokkot, összefésüljük őket és felváltva A-ba és B-be írjuk a rendezett 4 blokkot
 - A C-beli utolsó töredék blokk változatlanul A végére kerül

Külső rendezések

A

Rendezett: 4 blokk

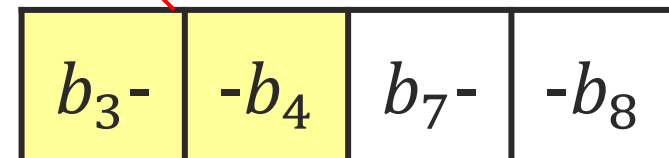
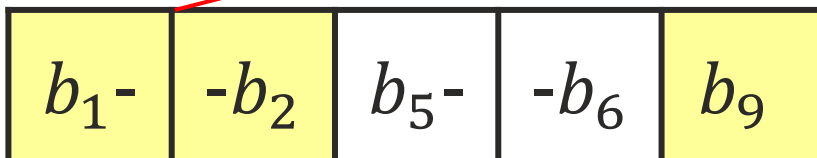
B



C

Összefuttatás
(RAM)

D



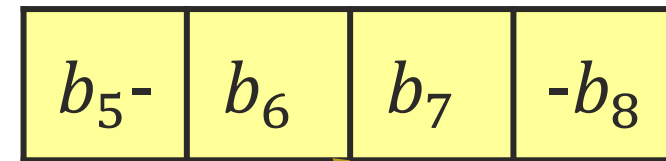
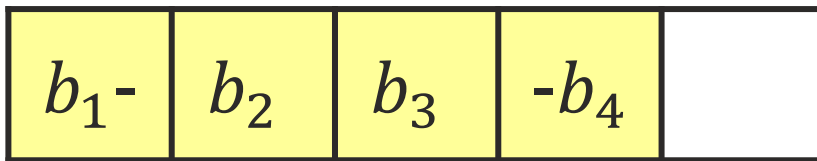
Rendezett: 2 blokk

Külső rendezések

A

Rendezett: 4 blokk

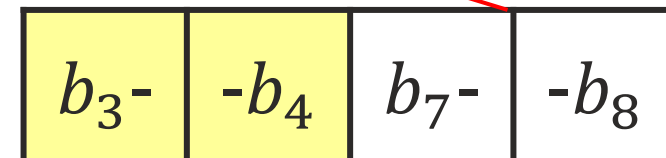
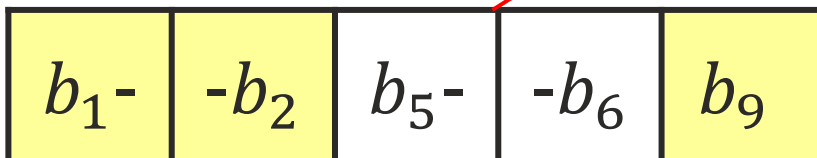
B



C

Összefuttatás
(RAM)

D



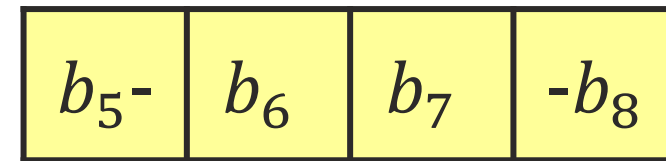
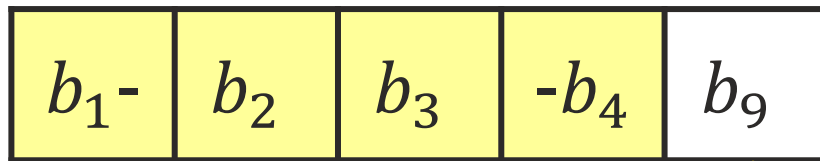
Rendezett: 2 blokk

Külső rendezések

A

Rendezett: 4 blokk

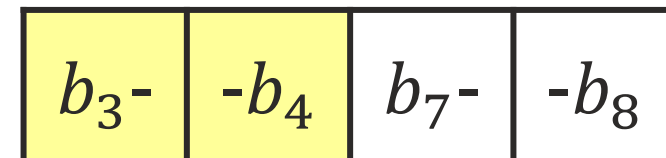
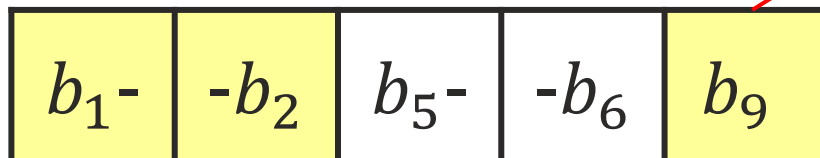
B



C

Összefuttatás
(RAM)

D



Rendezett: 2 blokk

Külső rendezések

- 4. menet
 - C-be kerül A és B 4-4 rendezett blokkja összefésülésének a 8 blokk hosszú rendezett eredménye, D-be pedig a maradék egy blokk.

Külső rendezések

A

Rendezett: 4 blokk

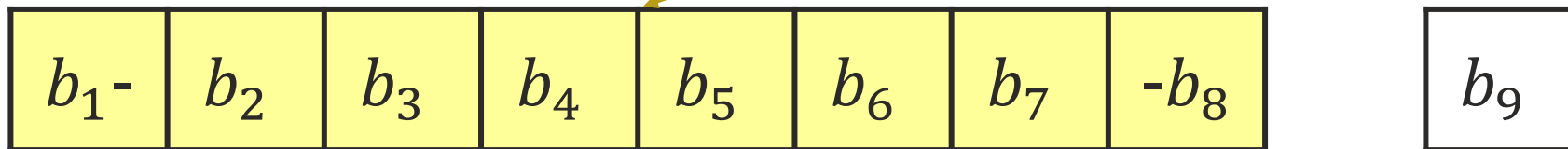
B



C

Összefuttatás
(RAM)

D



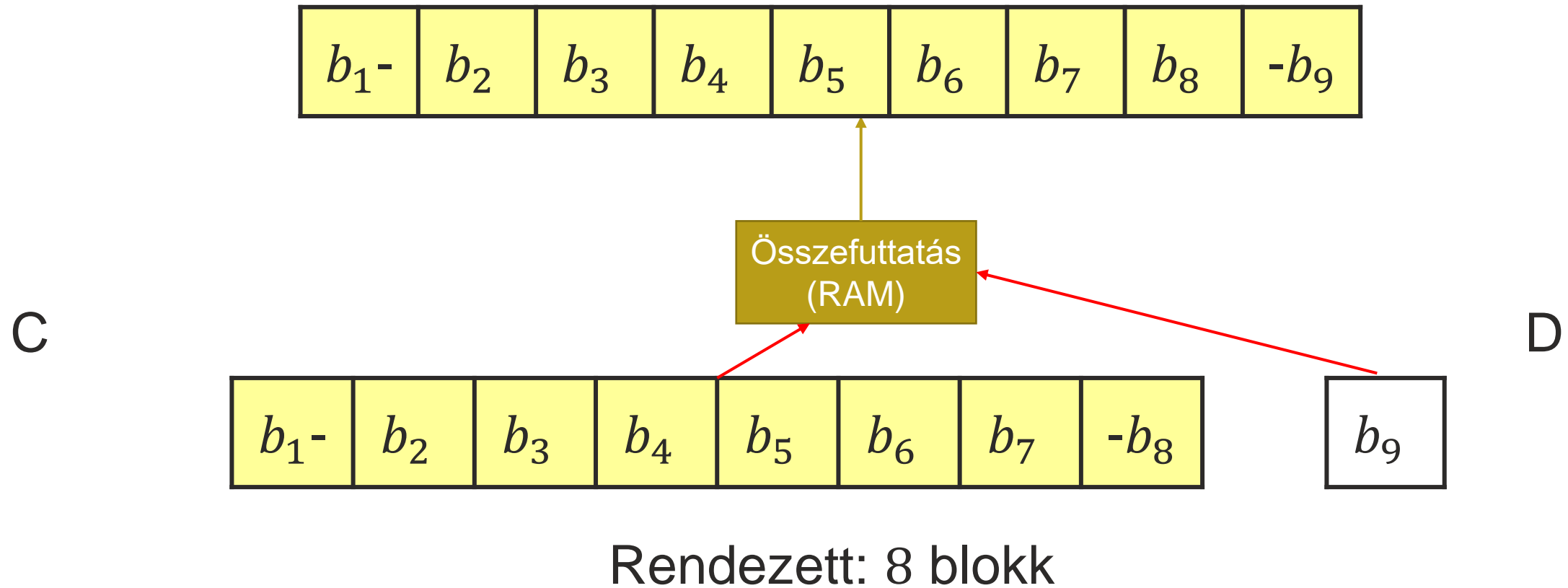
Rendezett: 8 blokk

Külső rendezések

- 5. menet
 - C 8 blokkját és D 1 blokkját összefésüljük S-be.
 - Elnevezés: egy k blokkból álló összefüggő rendezett részt **k hosszú futam**-nak nevezünk.

Külső rendezések

S – Rendezett



Külső rendezések

- Két kiegészítő megjegyzés:

1. Ebben a példában az S 9. blokkja csaknem végig nem került kapcsolatba más rendezett részekkel, csak az utolsó menetben került összefésülésre.
 - Ha végrehajtjuk a fenti eljárást $n = 15$ -re, akkor a páratlan töredék maradék rész mérete így alakul: 1, 3, 7, vagyis a fenti jelenség nem törvényszerű.
2. Ha a központi memória mérete korlátozott és nem képes befogadni az egyre növekvő méretű rendezett részeket (futamokat), akkor ezek összefésülését lehet „pufferelve”, akár blokkonként végezni.
 - Ez azért lehetséges, mert az összefuttatás egysége a rekord.

Külső rendezések

- Általában

- 1. menet eredménye: 1 hosszú futamok
- 2. menet eredménye: 2 hosszú futamok
- 3. menet eredménye: 4 hosszú futamok
-
- $(k - 1)$. menet eredménye: 2^{k-2} hosszú futamok
- k . (utolsó) menet eredménye: $\leq 2^{k-1}$ hosszú egyetlen futam = S
 - előtte maradék mindig lehet

Külső rendezések

- Az utolsó előtti $(k - 1)$. menetben még volt 2 futam
 - $2^{k-2} < n$
- A k . (utolsó) menetben már előáll a teljes rendezett fájl
 - $n \leq 2^{k-1}$
- Áttérve logaritmusra:
 - $k - 2 < \log_2 n \leq k - 1 \Rightarrow k - 1 = \lceil \log_2 n \rceil$
 - A menetek k száma: $k = \lceil \log_2 n \rceil + 1$
 - Az összes blokk-I/O száma: $2 * n * (\lceil \log_2 n \rceil + 1)$
 - minden menetben beolvastuk és kiírtuk mind az n blokkot

Külső rendezések

- Gyorsítási lehetőségek:
 - Nagyobb kezdő futamok létrehozása
 - Ha a központi memória lehetővé teszi, akkor az 1. menetben megtehetjük, hogy S -ből $m > 1$ blokkot olvasunk be, ezt rendezzük és az így keletkezett m hosszú kezdőfutamokat írjuk ki A -ba és B -be.
 - Ezután úgy megy tovább, hogy először két m hosszú futamokat fésülünk össze, majd két $2m$ hosszút, stb.

Külső rendezések

- Számítások

- 1. menet eredménye: m hosszú futamok
- 2. menet eredménye: $2 * m$ hosszú futamok
- 3. menet eredménye: $4 * m$ hosszú futamok
-
- $(k - 1)$. menet eredménye: $2^{k-2} * m$ hosszú futamok
- k . (utolsó) menet eredménye: $\leq 2^{k-1} * m$ hosszú egyetlen futam = S
- Innen $2^{k-2} * m < n \leq 2^{k-1} * m$
- A menetek k száma: $k = \left\lceil \log_2 \frac{n}{m} \right\rceil + 1$
- Az összes blokk-I/O száma: $2 * n * \left(\left\lceil \log_2 \frac{n}{m} \right\rceil + 1 \right)$

Külső rendezések

- Lehet azt is, hogy több, mint kétfelé fésülünk:
 - Ha az S fájl mellett nem $2 * 2$, hanem általában $2 * m$ fájlal dolgozunk, akkor ezzel a ráfordítással hatékonyabb eljárás nyerhető.
 - Legyen például $m = 3$, $n = 13$ és térjünk vissza az 1 hosszú kezdő futamokhoz:
 - Ekkor az 1. menetben felváltva A , B , C -be írjuk ki a rendezett kezdőfutamokat.
 - Ezután mindig három futamot fésülünk össze (a maradékoktól eltekintve) és az új futamokat felváltva A , B , C -be ill. D , E , F -be írjuk ki.
 - Az utolsó menetben S -be írjuk az eredményt.

Külső rendezések

S-ben $n = 13$

A B C

D E F

S

1. menet

1,1,1,1,1 1,1,1,1 1,1,1,1

2. menet

3,3, 3,1, 3

3. menet

9 4

4. menet

13

Külső rendezések

- A számolás

- 1. menet eredménye: 1 hosszú futamok
- 2. menet eredménye: m hosszú futamok
-
- $(k - 1)$ -edik menet eredménye: m^{k-2} hosszú futamok
- k -adik utolsó menet eredménye: $\leq m^{k-1}$ hosszú egyetlen futam=S
- Innen: $m^{k-2} < n$

$$k = \lceil \log_m n \rceil + 1 = \left\lceil \frac{\log_2 n}{\log_2 m} \right\rceil + 1$$

$$m^{k-1} \geq n$$

- Az összes blokk I/O művelet: $2n * \left(\left\lceil \frac{\log_2 n}{\log_2 m} \right\rceil + 1 \right)$

Külső rendezések

- Három fájlos külső rendező
 - Érdekes algoritmushoz jutunk, ha az m -felé fésülésnél nem $2m$ db fájlt használunk, hanem csak $m + 1$ db-t
 - Ezt $m = 2$ esetére nézzük meg, ami az eredeti eset. Ekkor tehát 3 fájlt használunk: A, B, C-t
 - Az első menetben szétosztjuk S immár rendezett blokkjait A-ba és B-be.
 - A második menetben elkezdjük A és B blokkjainak az összefésülését, de most csak 1 fájl tudja fogadni az eredményt, a C fájl. Ezért C-be fésülünk össze egészen addig, amíg A és B egyike ki nem ürül
 - Ekkor új menetet kezdünk a két nem üres fájllal
 - És így tovább

Külső rendezések

- Példa: $S_n = 13$ blokk

- A táblázatban az első szám a futamok számát jelenti, zárójelben pedig a futamok hossza áll

	1.	2.	3.	4.	5.	6.	7.	8.
A	7(1)	1(1)	0	1(5)	0	1(9)	0	1(13)
B	6(1)	0	1(3)	0	1(7)	0	1(11)	0
C		6(2)	5(2)	4(2)	3(2)	2(2)	1(2)	0

- Látjuk, hogy a második menet végén képződött 6 db 2 hosszú futam csak egyesével tud elfogyni – érezhetően sok lépésben
- Ennek oka az, hogy az 1. menet végi két futamszámnak 1 a különbsége
 - $7 - 6 = 1$
 - Hogy tudnánk ezen javítani?

Külső rendezések

- Eszünkbe jut a Fibonacci sorozat: 0,1,1,2,3,5,8,13,....
 - Ahol két szomszédos elem különbsége – az első néhány tag kivételével – 1-től különböző
 - Innen jön a gondolat, hogy az első menetben írjunk annyi futamot A-ba és B-be, mint a Fibonacci sorozat két (alkalmas) szomszédos eleme, és lépkedjünk visszafelé a sorozaton az egyes menetekben

	1.	2.	3.	4.	5.	6.
A	8(1)	3(1)	0	2(5)	1(5)	0
B	5(1)	0	3(3)	1(3)	0	1(13)
C		5(2)	2(2)	0	1(8)	0

Külső rendezések

- **Belátható**, hogy a 3 fájlos rendező éppen akkor fut le leggyorsabban, ha így járunk el, azaz A-ban és B-ben két szomszédos Fibonacci számnak megfelelő számú kezdő futamot hozunk létre.
 - Ha N nem Fibonacci szám, akkor vagy levágjuk és félretesszük a felesleget, és a végén még összefésüljük a kialakult eredménnyel, vagy pedig éppen fordítva: (virtuális) kiegészítéssel alkalmasan megnöveljük az input állomány méretét.

Külső rendezések

- Nézzük meg, hány lépésben érjük el F_k és F_{k-1} -ből az 1, 0 számokat úgy, hogy minden menetben egy újabb számot tudunk lefelé lépni

	1.	2.	...	$(k - 1).$	$k.$
$ S \approx F_{k+1}$	F_k	F_{k-1}	...	1	1
	F_{k-1}	F_{k-2}	...	1	0

- Látható, hogy a menetek száma ekkor k , hiszen F_k -től F_1 -ig vezet az út a táblázatban.
- Ki kell fejezni n -et F_{k+1} -gyel, ami az input fájl (közelítő) mérete blokkokban. Jelöljük $N = F_{k+1}$

Külső rendezések

- Számítás menete

- $N = F_{k+1}$
- $F_0 = 0, F_1 = 1, (F_2 = 1, F_3 = 2, F_4 = 3, \dots)$ $F_{k+1} = F_k + F_{k-1} (k \geq 1)$
- $F_k = \frac{1}{\sqrt{5}} * \left[\left(\frac{(1+\sqrt{5})}{2} \right)^k - \left(\frac{(1-\sqrt{5})}{2} \right)^k \right] \Rightarrow F_k \approx \frac{1}{\sqrt{5}} * \left(\frac{(1+\sqrt{5})}{2} \right)^k$
1,6180 – 0,6180
- Az $A = \frac{(1+\sqrt{5})}{2}$ az aranymetszés aránya, amely kielégíti az $A^2 - A - 1 = 0$ egyenletet.
- Ezt átrendezve: $A^2 = A + 1$, amiből teljes indukcióval megmutatható, hogy $k \geq 2$ esetén
 - $A^{k-2} \leq F_k \leq A^{k-1}$
 - Ha $N = F_{k+1}$, akkor $A^{k-1} \leq N$,
 - $k - 1 \leq \log_A N = \frac{\log_2 N}{\log_2 A} \approx 1,44 * \log_2 N$
 - $k \leq 1,44 * \log_2 N + 1$

2-3 fa / B-fa

Következő téma