ADATSZERKEZETEK ÉS ALGORITMUSOK

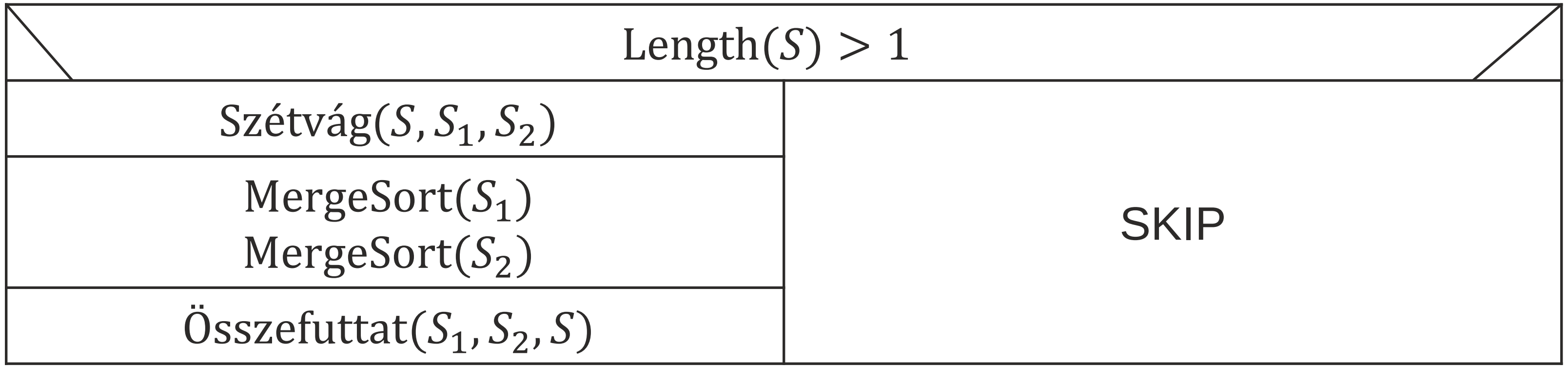
Batcher-féle rendező

Batcher-féle rendezés

- Batcher-féle páros-páratlan összefésüléses rendezés
- Jelentősége: ez a MergeSort olyan változata, amelyben a lépések jelentős része párhuzamosan végezhető el.
  - Ha párhuzamos processzorokon hajtanánk végre, akkor azt tapasztalnánk, hogy $\Theta((\log_2 n)^2)$ idő alatt futna le, szemben a MergeSort $\Theta(n * \log_2 n)$ idejével

Összefuttatásos rendezés

- $\text{MergeSort}(S)$



Összefuttatásos rendezés – tömbre

- $\mathrm{MergeSort}(A, k, v)$

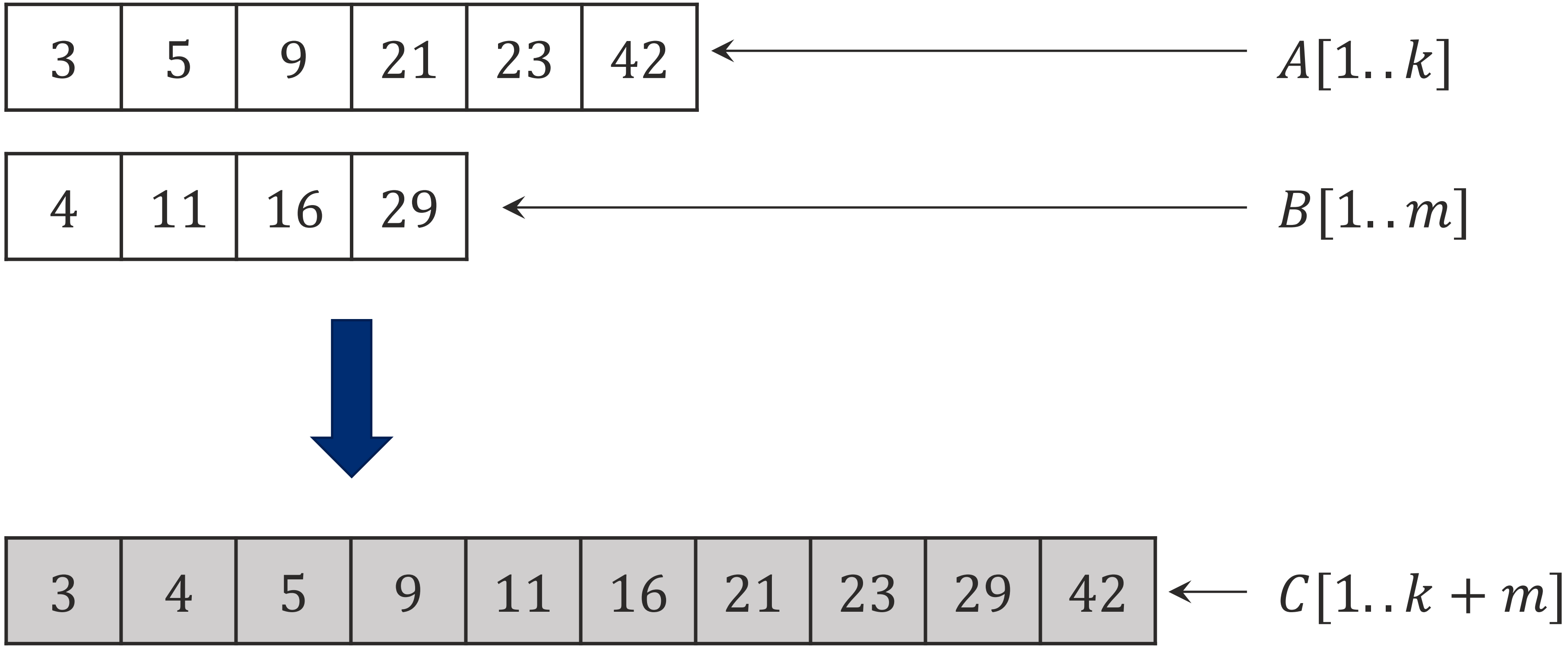
<table>
<tr><td colspan="2">$k < v$</td></tr>
<tr><td>$h \leftarrow \frac{k+v}{2}$
$\mathrm{Szétvág}(S, S_1, S_2)$</td><td rowspan="3">SKIP</td></tr>
<tr><td>$\mathrm{MergeSort}(A, k, h)$
$\mathrm{MergeSort}(A, h+1, v)$</td></tr>
<tr><td>$\mathrm{Összefuttat}(A[k, h], A[h+1, v], A[k, v])$</td></tr>
</table>

- A külső hívás
  - $\mathrm{MergeSort}(A, 1, n)$
- Az összefuttatásnál szükséges egy segédtömb használata

Mergesort

- Hatékonyságelemzés:
- Műveletigény:
  - Ha n a két sorozat együttes hossza:
    - $\text{MÖö}_{\text{sszefuttat}}(n) = n - 1$
    - $\text{MÖ}_{\text{MS}}(n) \leq (n - 1) * \log_2 n = \Theta(n \log_2 n)$

Mergesort



Mergesort

$$A = a_1 < a_2 < \cdots < a_k$$

$$B = b_1 < b_2 < \cdots < b_m$$

$$\Rightarrow$$

$$C = c_1 < c_2 < \cdots < c_{m+k}$$

Batcher-féle

- Ez lesz a PPMerge
  - Egy tömb(részlet) két rendezett feléből összefésüléssel előállítja a tömb(részlet) rendezett tartalmát
  - Ez is rekurzív eljárás, csak kicsit másképp

Batcher-féle rendezés

- Az új összefésülés:
  1. A páratlan és B páros indexű elemeit fésüli össze U-ba
  2. A páros és B páratlan indexű elemeit fésüli össze V-be

$$A = a_1 < a_3 < a_5 < \cdots$$

$$B = b_2 < b_4 < b_6 < \cdots$$

$$\Rightarrow U = u_1 < u_2 < u_3 < u_4 < \cdots$$

$$A = a_2 < a_4 < a_6 < \cdots$$

$$B = b_1 < b_3 < b_5 < \cdots$$

$$\Rightarrow V = v_1 < v_2 < v_3 < v_4 < \cdots$$

- Tegyük fel, hogy $k = m$ vagy $k = m + 1$

Batcher-féle rendezés

3. Az U és V sorozatokat nem kell összefésülni, hanem elég csak az egymás alatti u_i, v_i párokat 1-1 összehasonlítással a helyes sorrendbe rakni
   - Így $\{u_1, v_1\}, \{u_2, v_2\}, \{u_3, v_3\}, \ldots$ -ból kialakul a helyes sorrend (Ezt be fogjuk látni.)
   - Amennyiben az U és V sorozat hossza eltérő (tehát az végeredmény páratlan hosszú), az utolsó elemet összehasonlítás nélkül kell áttenni.
     - Ezt az esetet külön kell vizsgálni.

- Párhuzamosítás: 1. és 2. párhuzamosan végezhető.

Batcher-féle rendezés – példa

A	1	4	5	7	11	12	14	20
B	2	3	6	10	13	15	16	17

- 1. menet: A páratlan és B páros indexű elemei

A	1	5	11	14
B	3	10	15	17

U	1	3	5	10	11	14	15	17

Batcher-féle rendezés – példa

A	1	4	5	7	11	12	14	20
B	2	3	6	10	13	15	16	17

- 2. menet: A páros és B páratlan indexű elemei

A	4	7	12	20
B	2	6	13	16

V	2	4	6	7	12	13	16	20

Batcher-féle rendezés

- $\{u_1, v_1\}, \{u_2, v_2\}, \{u_3, v_3\}$ párosítás

U	1	3	5	10	11	14	15	17
V	2	4	6	7	12	13	16	20

1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20

Batcher-féle rendezés

- Rekurzió
  - A 2-2 rövidebb sorozat összefésülését ugyanezzel az eljárással végezzük. Ez rekurzív hívással valósul meg.
  - Ehhez meg kell adni a legkisebb k és m értéket, amelyre már nem hívja az eljárás önmagát:
    - Egy 2 hosszú tömböt még szétvágunk két 1 hosszú részre és azokra meghívjuk az összefésülőt, ekkor
      $k = 1$ és $m = 1$.
    - Egy 1 hosszú tömböt azonban már nem fésülünk össze, hanem felismerjük, hogy az már rendezett, így nulla (0) összehasonlítást igényel, ekkor $k = 1$, $m = 0$.

Batcher-féle rendezés

- Állítás: A PP_Merge eljárás helyes.
  - Belátjuk, hogy a leírt eljárás azt a helyes rendezett sorozatot eredményezi, amelyet $C = c_1 < c_2 < \cdots < c_{m+k}$-val jelöltünk
  - Esetszétválasztással gondoljuk meg:
  $$c_1 = \min\{u_1, v_1\}, \qquad c_2 = \max\{u_1, v_1\}$$
  - Általában, ha $1 \leq i \leq \left\lfloor \frac{k+m}{2} \right\rfloor$:
  $$c_{2i-1} = \min\{u_i, v_i\}, \qquad c_{2i} = \max\{u_i, v_i\}$$

Batcher-féle rendezés

- Vegyük figyelembe a kiegyensúlyozott szétvágást is, azaz azt, hogy $k = m$ vagy $k = m + 1$ esetén $\left\lfloor \frac{k+m}{2} \right\rfloor = m$
- Ha $k + m$ páratlan, akkor még azt is be kell látni, hogy c_{k+m} is helyesen képződik, hiszen erre akkor a képlet nem vonatkozik.
- Bizonyítás:
  1. Belátjuk, hogy a C sorozatban bármely páros indexű tagig elmenve $c_1, \dots c_{2k}$ között ugyanannyi u_i szerepel, mint v_j vagyis az U és a V sorozat szinte „cipzár”-szerűen építi a C-t.

Batcher-féle rendezés

- A C sorozat a rendezett A és B sorozatok összefésülésével adódik.
- Tegyük fel, hogy
  - A-ból az $a_1, \dots a_s$ elemek,
  - B-ből a $b_1, \dots, b_{2k-s}$ elemek jönnek összefésüléssel a
  - C sorozat első $2k$ elemébe

Batcher-féle rendezés

$$\{c_1, \dots\ c_{2k}\} = \{a_1, \dots a_s\} \cup \{b_1, \dots\ b_{2k-s}\}$$

U-ba kerülnek a páratlan indexű elemek, ezek száma: $\lceil s/2 \rceil$	V-be kerülnek a páros indexű elemek, ezek száma: $\lfloor s/2 \rfloor$	U-ba kerülnek a páros indexű elemek, ezek száma: $\lfloor (2k-s)/2 \rfloor$	V-be kerülnek a páratlan indexű elemek, ezek száma: $\lceil (2k-s)/2 \rceil$

Batcher-féle rendezés

- Az állítás ekkor egyenértékű azzal, hogy:
- $\lceil s/2 \rceil + \lfloor (2k-s)/2 \rfloor = \lfloor s/2 \rfloor + \lceil (2k-s)/2 \rceil$?
  - Ez nyilván igaz, ha s páros, hiszen ekkor minden tag egész,
  - Ha s páratlan, akkor a kérdés a következő egyenlőség fennáll-e (igen)
  - $\frac{s+1}{2} + \frac{2k-(s+1)}{2} = \frac{(s-1)}{2} + \frac{2k-(s-1)}{2}$

Batcher-féle rendezés

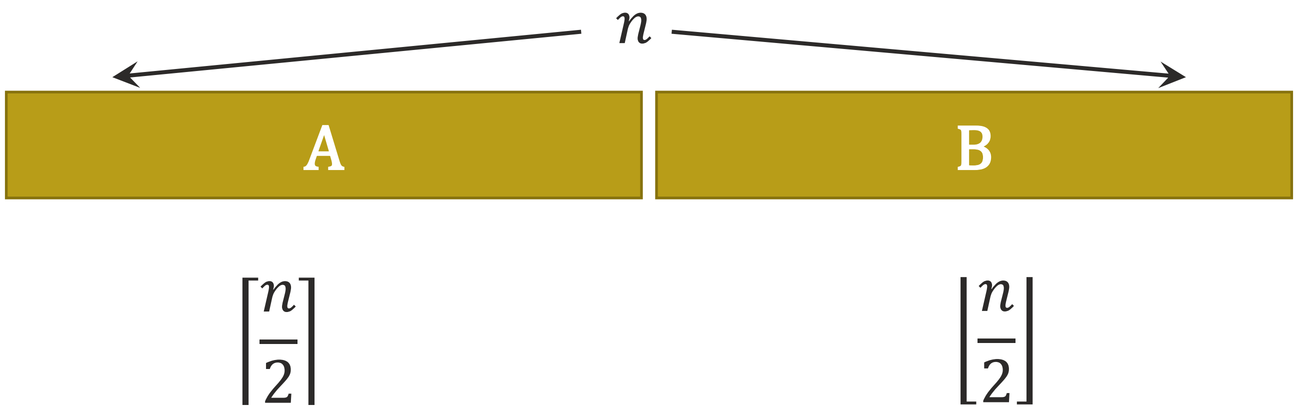
2. Eszerint:
    - $\{c_1, \dots\ c_{2i}\} = \{u_1, \dots u_i\} \cup \{v_1, \dots v_i\}$
    - $\{c_1, \dots\ c_{2(i-1)}\} = \{u_1, \dots u_{i-1}\} \cup \{v_1, \dots v_{i-1}\}$
    - A két halmaz kivonásával:
        - $\{c_{2i-1}, c_{2i}\} = \{u_i, v_i\}$
    - Így, mivel $c_{2i-1} < c_{2i}$, tehát az eredeti állítás fennáll.

- Tehát
    - U és V már egy párhuzamos lépésben összefésülhető.

Batcher-féle rendezés

- Hatékonyságelemzés:
  - Párhuzamos költséget számolunk, ez azt jelenti, hogy akárhány összehasonlítás is csak 1-nek számít, ha párhuzamosan végezzük!

1. A PP_Merge rekurzív eljárást egy n méretű, két (közel) egyenlő méretű, rendezett két félből álló tömbre hajtjuk végre.



Batcher-féle rendezés

- A két félből párhuzamosan lehet képezni az U és V sorozatokat, itt tehát $\lceil n/2 \rceil$-t vesszük alapul. Mivel ugyanezt az eljárást használjuk az U és V előállítására is, ez a költségszámítás képletében is rekurziót ad.
- Továbbá, az U és V tömbökből egyetlen párhuzamos összehasonlítással kapjuk a C eredményt.
  - Így az egyenlet: $\mathrm{MÖ}_{\mathrm{PPM}}(n) \leq \mathrm{MÖ}_{\mathrm{PPM}}\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1$
  - Itt egyébként elegendő MÖ helyett Ö-t írni, mert ez az eljárás fix összehasonlítás számmal dolgozik. Szokásos jelölés még:
    - $T(n) \leq T\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1, T(1) = 0$

Batcher-féle rendezés

- Ennek megoldása úgy történik általánosan is, ahogy egy konkrét értékre:
  - például $n = 21$-re:
  - $T(21) \leq T(11) + 1 \leq T(6) + 1 + 1 \leq T(3) + 1 + 1 + 1$
    $\leq T(2) + 1 + 1 + 1 + 1 \leq T(1) + 1 + 1 + 1 + 1 + 1 =$
    $= 0 + 1 + 1 + 1 + 1 + 1 = 5 = \lceil \log_2 21 \rceil$
  - $T(21) \leq \lceil \log_2 21 \rceil$
  - Ha v olyan egész, amire $2^v < n \leq 2^{v+1}$, akkor a rekurzív egyenletet éppen $v + 1$-szer alkalmazzuk, mire eljutunk
    $T(1)$-ig
    - Így $v + 1$ db egyest adunk össze, vagyis $T(n) \leq \lceil \log_2 n \rceil$
  - Ez tehát a párhuzamos összefésülés költsége.

Batcher-féle rendezés

2. A párhuzamos rendező eljárás szerkezete pontosan az, ami a bevezetőben felidézett MergeSort-é, csak más összefuttatást alkalmaz.

Batcher-féle rendezés

2. A párhuzamos összehasonlítás számra az egyenlet:

- $\text{Ö}_{\text{Batcher}(n)} \leq \text{Ö}_{\text{Batcher}}(\lceil n/2 \rceil) + \lceil \log_2 n \rceil$
- Egyszerűbb jelöléssel:
  - $T(n) \leq \ T(\lceil n/2 \rceil) + \lceil \log_2 n \rceil$
- Ha ezt is kifejtjük, akkor az előzőhöz hasonlóan azt kapjuk, hogy $\lceil \log_2 n \rceil$ számú tag lesz; ezek a tagok azonban csökkennek:
  - $T(n) \leq T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \lceil \log_2 n \rceil \leq$

$$T\left(\left\lceil \frac{n}{2^2} \right\rceil\right) + \left\lceil \log_2 \left(\left\lceil \frac{n}{2} \right\rceil\right)\right\rceil + \lceil \log_2 n \rceil \leq \cdots$$

$$\leq T(1) + 1 + 2 + \cdots \ (\lceil \log_2 n \rceil - 1) + \lceil \log_2 n \rceil =$$

$$\frac{(\lceil \log_2 n \rceil + 1) * \lceil \log_2 n \rceil}{2} \approx \frac{(\log_2 n)^2}{2} = \Theta(\log_2 n)^2$$

Batcher-féle rendezés

- Az előzőekben felhasználtuk, hogy
- $\lceil\log_2(\lceil n/2\rceil)\rceil = \lceil\log_2 n\rceil - 1$, ezt be is látjuk:
  - Ha $n = 2p$, akkor igaz.
  - Ha $2p < n \leq 2p + 1$, akkor
  - $2p - 1 < \frac{n}{2} \leq 2p$,
  - $2p - 1 < \left\lceil\frac{n}{2}\right\rceil \leq 2p$
  - $p - 1 < \log_2\left\lceil\frac{n}{2}\right\rceil \leq\ p$, így
  - $\left\lceil\log_2\left(\left\lceil\frac{n}{2}\right\rceil\right)\right\rceil =\ \lceil\log_2 n\rceil - 1$

Batcher-féle rendezés - tömbökre

- BatcherSort(S, p, q)

<table>
<tr><td colspan="2">$p < q$</td></tr>
<tr><td>$r \leftarrow \lfloor (p+q)/2 \rfloor$</td><td rowspan="3">SKIP</td></tr>
<tr><td>$\mathrm{BatcherSort}(S, p, r)$
$\mathrm{BatcherSort}(S, r+1, q)$</td></tr>
<tr><td>$\mathrm{PP\_Merge}(S[p \dots r], S[r+1, \dots q], S[p, \dots q])$</td></tr>
</table>

- Külső hívása
    - BatcherSort(S,1,n)

Rendezők implementációja

Következő téma