Sammansmältningssortering
Weiss, avsnitt 7.6

Peter Ljunglöf

DAT036, Datastrukturer

3 dec 2012

Merge

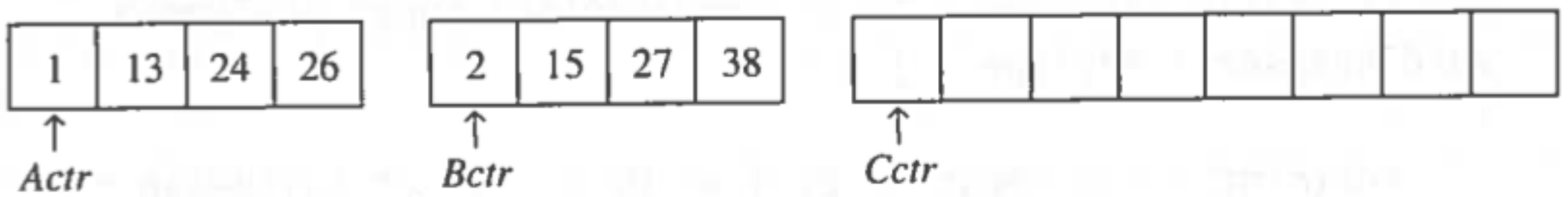
Den grundläggande komponenten i mergesort är själva sammanflätningen.

- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!

Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

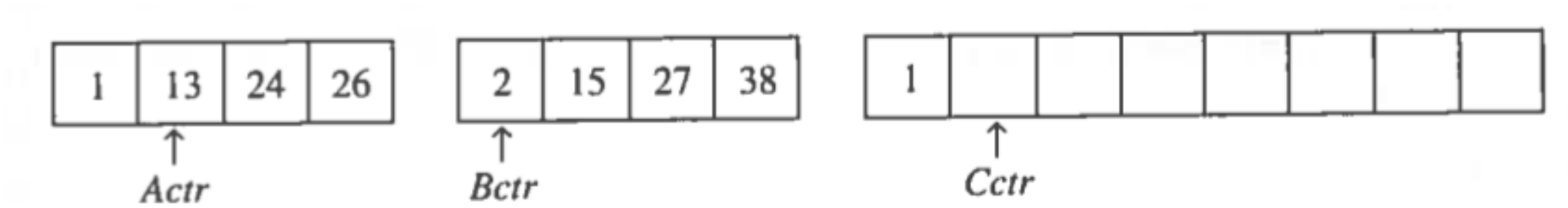
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

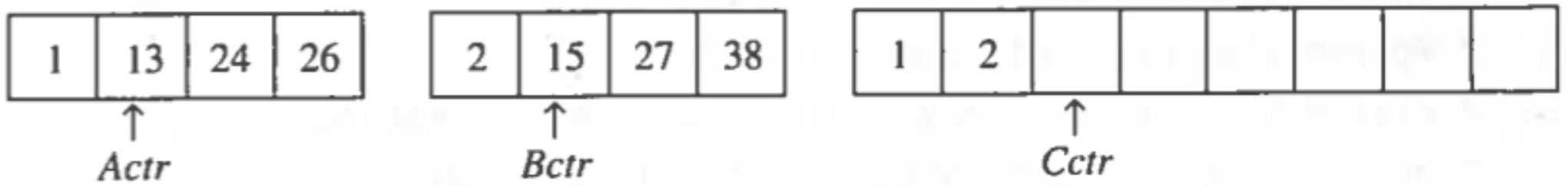
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

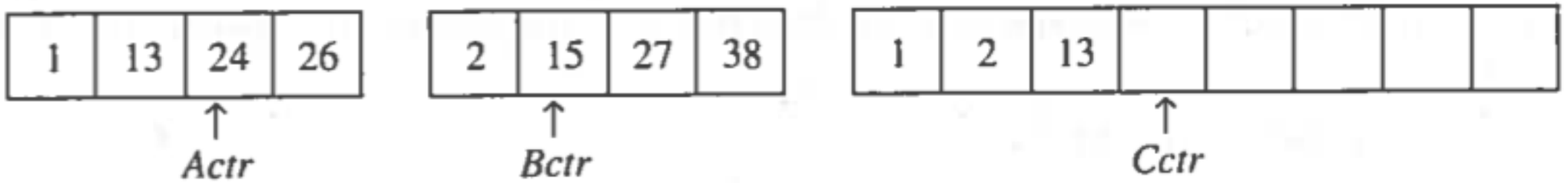
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

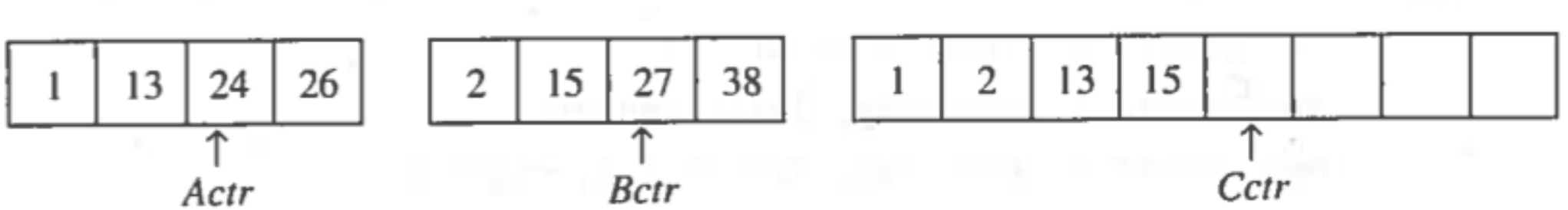
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

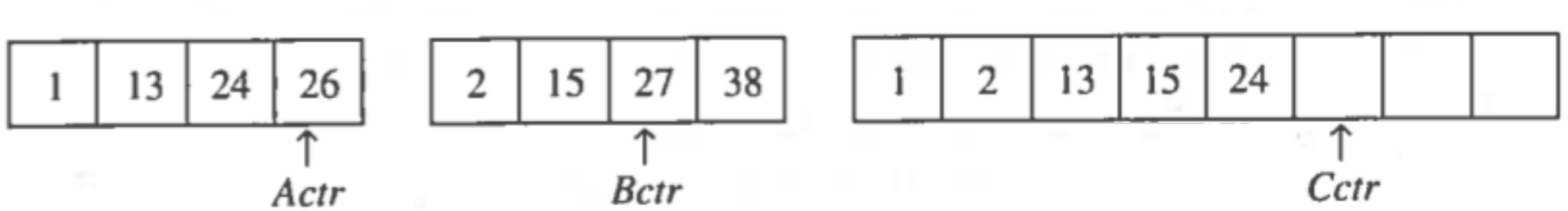
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

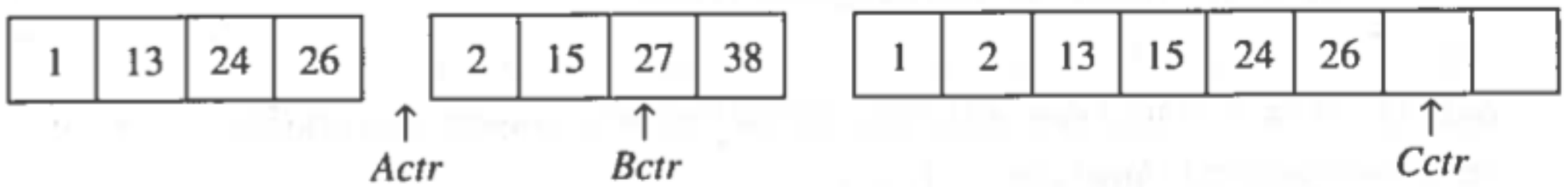
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

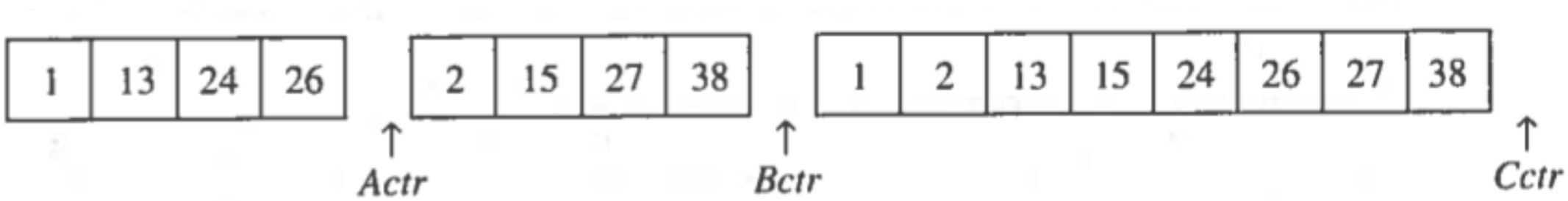
- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Merge

Den grundläggande komponenten i mergesort är själva sammanflätningen.

- den tar två listor och flätar ihop dem
- kravet är att bägge listorna är sorterade!



Analys av Merge

Varje jämförelse flyttar ett element från någon av listorna till den sammanslagna listan:

- alltså totalt $n+m$ jämförelser
- komplexitet $O(n)$, om vi antar att n är längden på den längsta listan

Merge sort

Merge kan användas som en del i en rekursiv sorteringsalgorim:

- dela upp listan i två halvor (split)
- sortera den ena halvan (rekursivt)
- sortera den andra halvan (rekursivt)
- slå ihop de sorterade halvorna (merge)

Merge sort, rekursiv algoritm

metod mergesort(*list*):

- om *list* innehåller mer än ett element:
  - skapa två listor *left* + *right*
  - kopiera den första halvan av *list* till *left*
  - kopiera den andra halvan av *list* till *right*
  - mergesort(*left*)
  - mergesort(*right*)
  - merge(*left*, *right*, *list*)
  - fläta samman *left* + *right* och stoppa i *list*

Komplexitetsanalys

Varje ”nivå” innehåller n element:

- för att dela upp respektive slå ihop en nivå krävs $O(n)$ steg

Varje ”nivå” innehåller dubbelt så många del-listor som den föregående

- ända tills del-listorna bara har 1 element
- alltså finns det $O(\log n)$ nivåer

Komplexiteten blir alltså

- $O(n) \cdot O(\log n) = O(n \log n)$