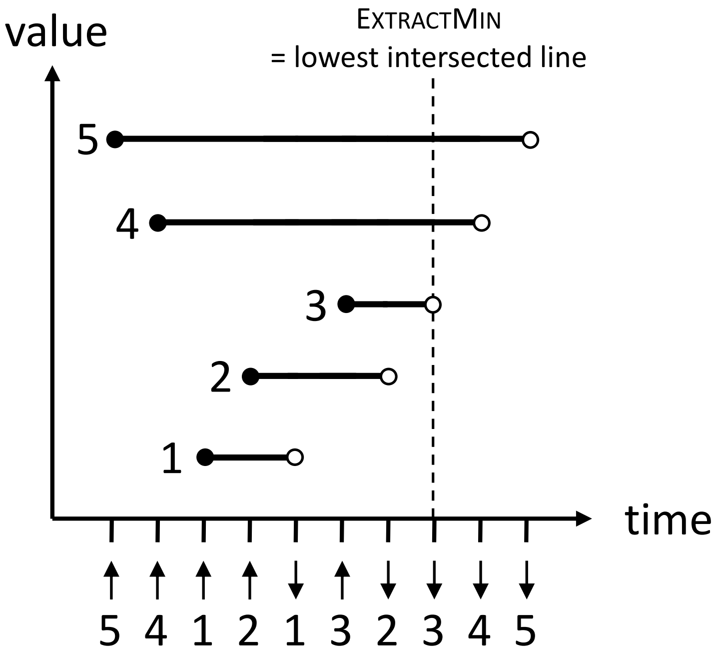
Soft Sequence Heaps

Gerth Stølting Brodal

Aarhus University

SIAM Symposium on Simplicity in Algorithms (SOSA), January 11, 2021

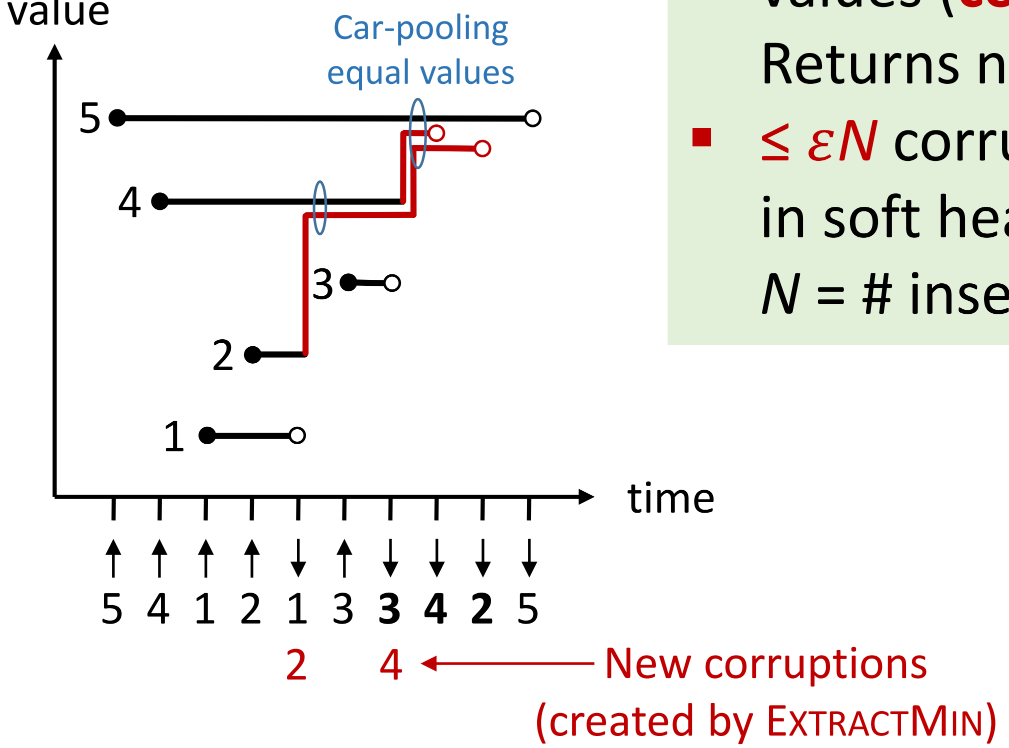
Heap



↑ Insert(x)

↓ ExtractMin()

Soft Heap



Soft heap properties

- ExtractMin can increase values (**corruptions**)
  Returns new corruptions
- $\leq \varepsilon N$ corrupted elements in soft heap, $0 \leq \varepsilon \leq ½$, N = # insertions

(other operations not discussed in this talk MakeHeap, Meld, FindMin, Delete)

Soft heap results

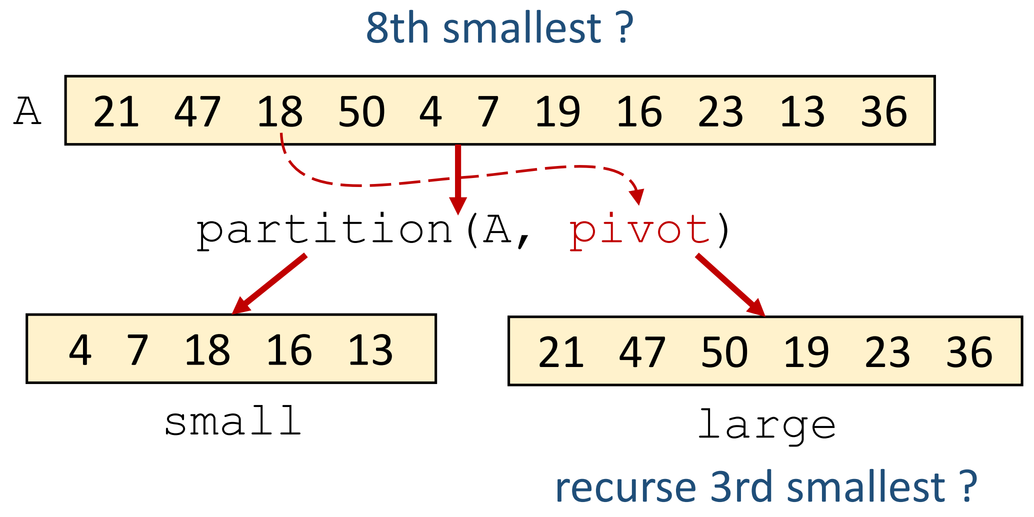
Soft heaps	INSERT / EXTRACTMIN	Reference	Applications
Introduced car-pooling Binomial trees	O(log $\frac{1}{\varepsilon}$) / O(1)	Chazelle ESA98*/JACM00 *2018 ESA Test-of-Time award	Selection MST O(m·α(m, n))
		Pettie, Ramachandran JACM02	MST optimal Unknown complexity
“A simpler … soft heaps” Balanced binary trees	O(log $\frac{1}{\varepsilon}$) / O(1)	Kaplan, Zwick SODA09	
“Soft heaps simplified” Balanced binary trees	O(1) / O(log $\frac{1}{\varepsilon}$)	Kaplan, Tarjan, Zwick SICOMP13	
Report corruptions Tag corrupted reported items Corruptions only EXTRACTMIN	O(1) / O(log $\frac{1}{\varepsilon}$)	Kaplan, Kozma, Zamir, Zwick SOSA19	Heap selection (and related) Simplifying Frederickson JCSS93
Soft sequence heaps	O(log $\frac{1}{\varepsilon}$) / O(1)	Brodal SOSA21	

Time bounds are all amortized

Application of Soft Heaps – O(n) Selection

```
function select(A, k)
  if k = 1 then
     return min(A)
  Q = softheap(1/3)
  for a ∈ A do
     Q.INSERT(a)
  for i = 1 to |A|/3 do
     pivot = Q.EXTRACTMIN()
  small, large = partition(A, pivot)
  if k ≤ |small| then
     return select(small, k)
  return select(large, k - |small|)
```

use soft heap to find pivot



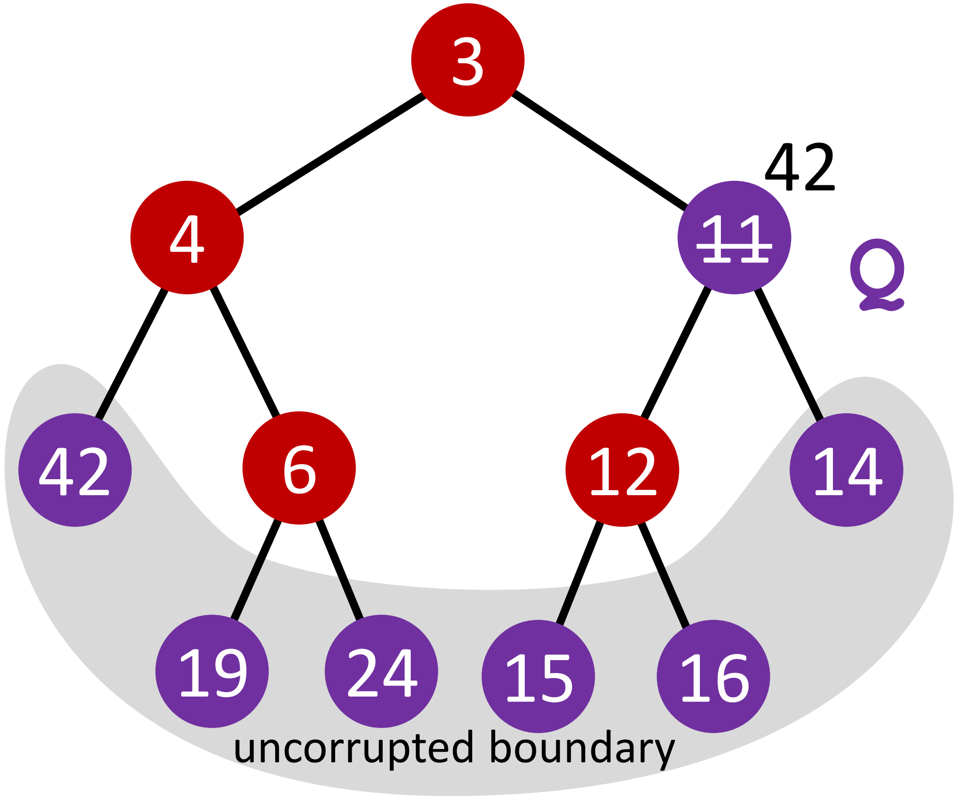
pivot rank $\in \left[\frac{|A|}{3}, \frac{2|A|}{3}\right]$
(pivot is the increased value)

$T(n) \leq T(2n/3) + O(n)$

Chazelle JACM00

Application of Soft Heaps – O(*k*) Heap Selection

```
function select(root, k)
  S = {root}
  Q = softheap(1/4)
  Q.INSERT(root)
  for i = 1 to k - 1 do
     (e, C) = Q.EXTRACTMIN()
     if e not corrupted then
        C = C U {e}
     for e ∈ C do
       Q.INSERT(e.left)
       Q.INSERT(e.right)
       S = S U {e.left, e.right}
  return select(S, k)
```

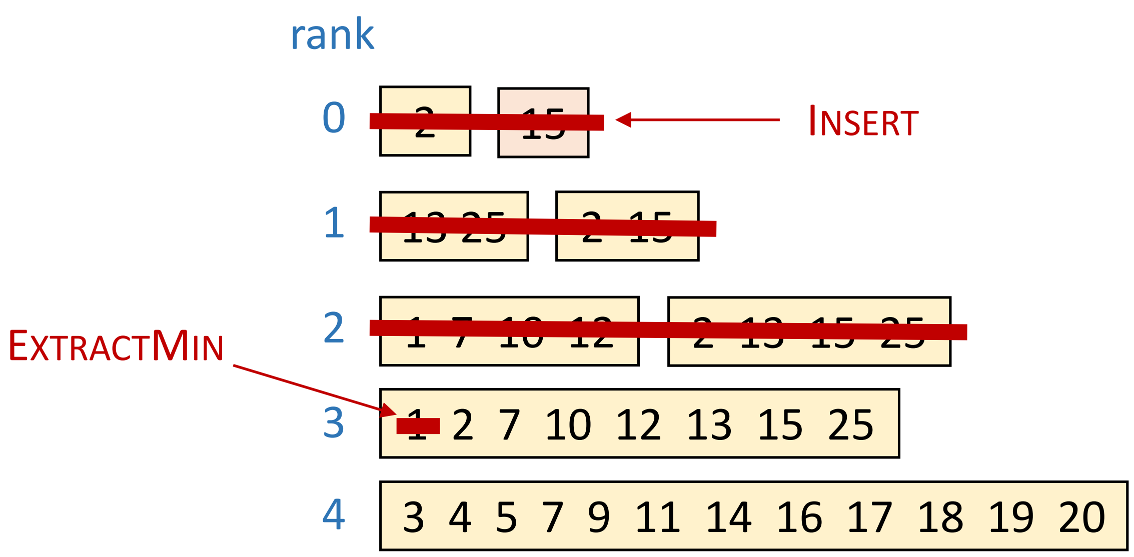


Kaplan, Kozma, Zamir, Zwick SOSA19

Sequence Heaps

Sequence heap properties

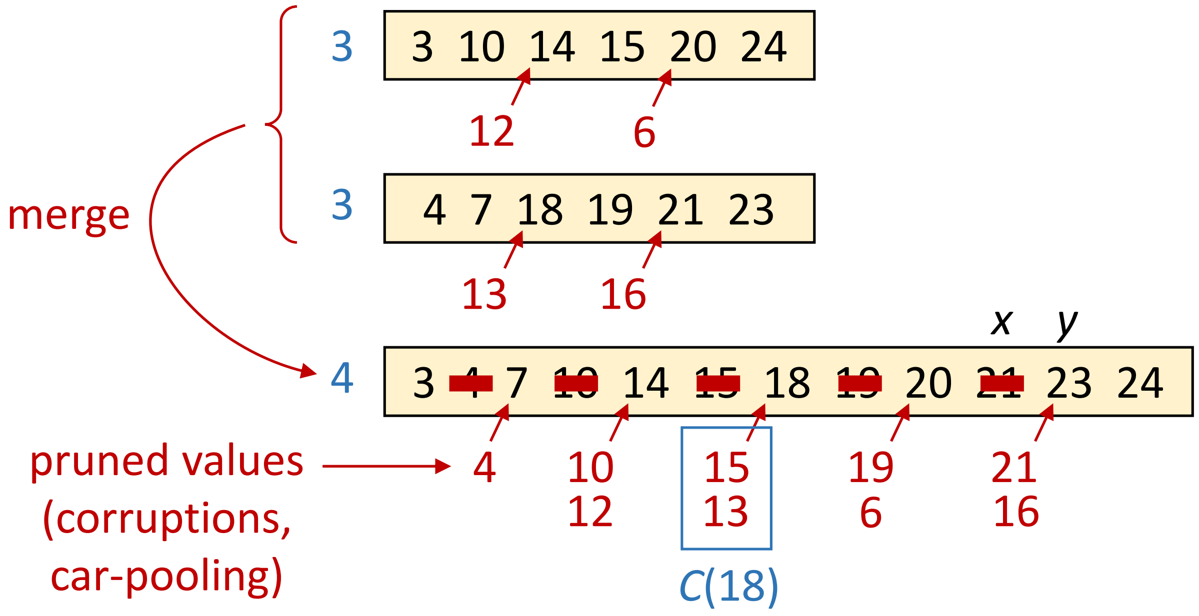
- Sorted lists, each list a rank
- Two lists rank $r \Longrightarrow$ **merge**, rank r+1
- Rank r list ≤ 2^r values
- N INSERT $\Longrightarrow$ rank ≤ log N
- INSERT and EXTRACTMIN amortized O(log N)




```
INSERT(x)
   Create rank 0 list containing x
   while two list have equal rank do
     merge the two lists

EXTRAXTMIN()
   Find list with smallest head element e
   Remove and return e
```

Soft Sequence Heap



Soft Sequence Heap properties

- Sorted lists, each list a rank
- Prune every 2nd element of a new list of even rank > $\log \frac{1}{\varepsilon}$
- x pruned $\Longrightarrow \{ x \} \cup C(x)$ added to $C(y)$ where y successor of x
- Rank r list $\Longrightarrow$ size $\leq \sqrt{2^r/\varepsilon}$

```
INSERT(x)
   Create rank 0 list containing x
   while two list have equal rank r do
     merge the two lists
     if r even and r > log 1/ε then
        prune list

EXTRAXTMIN()
   Find list with smallest head element e
   if |C(x)| = 0 then
     Remove and return e
   else
     Remove and return an element from C(e)
```

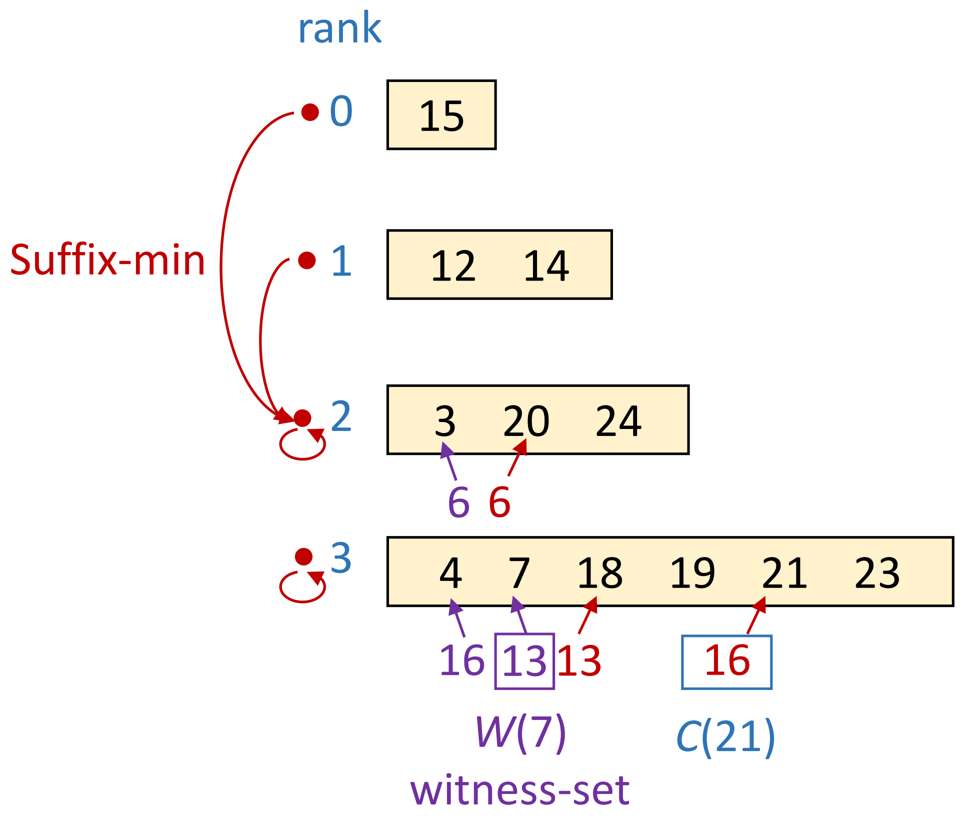
How can this work ?

- Only $O(\sqrt{n/\varepsilon})$ elements are not pruned

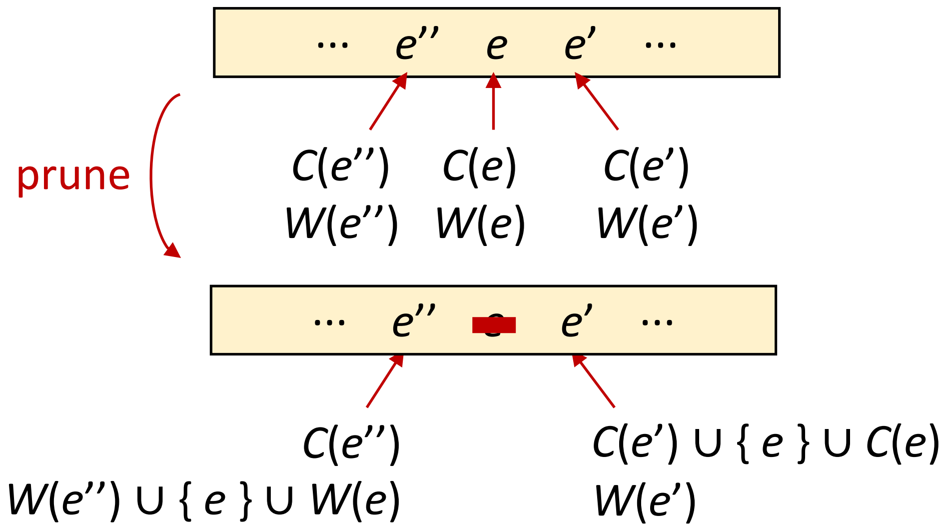
Solution

- Not all pruned elements need to be considered corrupted

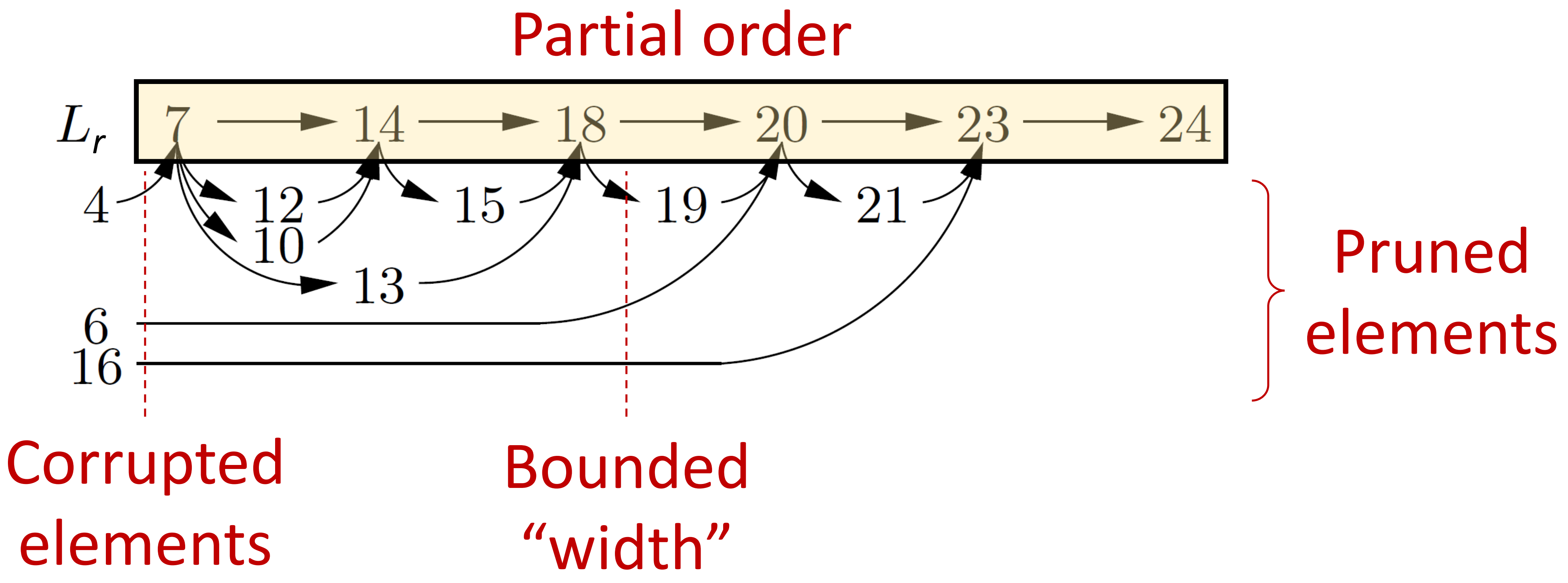
Suffix-min pointers and witness sets



- Each non-pruned element has a corruption set $C(e)$ and witness-set $W(e)$
- $x \in C(e) \Rightarrow x \leq e$
  $x \in W(e) \Rightarrow e \leq x$
- x corrupted $\Leftrightarrow x \in C(e')$ for some e' and $x \notin W(e'')$ for any e''
- When ExtractMin removes e, $W(e)$ is reported as corrupted



Analysis – corruptions ≤ εn



- $C(e)$ doubles when pruning $\Longrightarrow |C(e)| \le 2^{(r - \log\frac{1}{\varepsilon})/2} \le \sqrt{2^r/\varepsilon}$
- "width" doubles when merging + increases by $\sqrt{2^r/\varepsilon}$ when pruning $\Longrightarrow$ "width" ≤ εn

Summary - Soft Sequence Heaps

- At most εn corruptions in heap
- INSERT and EXTRACTMIN amortized time $O(\log \frac{1}{\varepsilon})$
- Witness-sets used in analysis and for reporting corruptions
  - can be removed from construction if reporting not needed
  - only $\sqrt{n/\varepsilon}$ elements are not in corruption sets (previous constructions $\Theta(n)$)

Further results in paper

- Discuss how to remove buffering insertions from previous constructions

Open problems

- I/O & cache oblivious soft heaps with $O(B)$ operations taking $O(1)$ I/Os ?
- Other applications of soft heaps ?