

# External Priority Queue

[Fall of 2019, 99]

Min:  $T_{insert} : O_A \left( \frac{1}{B} \cdot \log_{N/B} N/B \right)$   
 $D_{min} : \frac{1}{B} \cdot \log_{N/B} N/B$

(22)

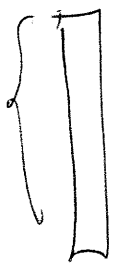
Plds :  $O(N/B)$

[CPU :  $O(\log_2 N)$  for ops.]

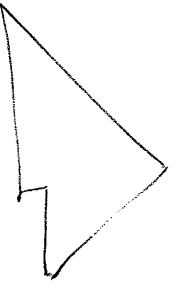
$\Rightarrow$  Efficient alternative HeapSort ( $= N \cdot T_{insert} + N \cdot D_{min}$ )

Remark for same method :  $B = P$   
 $Min = Max$  (max heap)

CRAM :  
 Insert buffer  
 (an internal heap)  
 $H' = \Theta(N)$  plds. (eg  $H' = N/2$ )



Tree of fanout  $\Theta(N/B)$   
 Shaps:



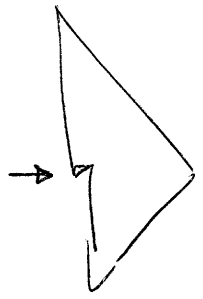
Nodes :  $\underbrace{\quad\quad\quad}_{\text{unsorted}} - \underbrace{\quad\quad\quad}_{\text{sorted}}$

$\Theta(N/B)$  plds

Heap ordered : Plds in nodes buffer dominates any plds in subtree.

Up to  $H'$  plds in unsorted buffer

Last node:



- Part <sup>is the</sup>  $V$  only interval node of forest  $< N/2$
- Node buffer may contain  $\leq N/2$  nodes.

All other nodes: Buffer contains at least  $N/2$  nodes and at most  $n$  nodes.

Operations

Insert into circuit buffer (heap in RAM)

If size  $\geq N$  ~~insert into circuit buffer~~

Lowest state load

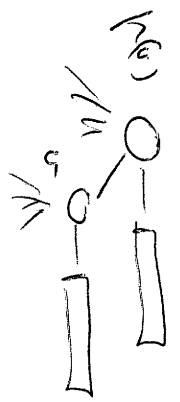
First input buffer also filling backdoor

node buffer (sorter forest)

Flip node id, state load

Shift Up or left ~~if needed~~ (if needed).

# Split Up



← Some elements in  
 can-fields (heap-order-  
 wise) with ancestors.  
 Bst tree OK.

[Note: can be checked by  
 comparing strongest elem. in  
 u with weakest in p(u)].

Merge the two (sorted) node buffers.

Under (array?) Split into same sized buffers as before.  
 started in → sorted outputs

think you  
 its

Note: Heap order restored between u and p(u).

Heap order between p(u) and p(p(u))  
 may be violated. ⇒ recurse SplitUp on

p(u).

# Disaster Best:

Check rest of tree [rest block of its buffer length in  $TRM$ ],  
and insert buffer (a heap).

IF best is in inset buffer, remove & return.

Else remove from root node buffer

IF root buffer  $< n/2$ :

## Refill (root).

### Clean Up leaves

## Refill (n)

Do  $n/2$  wangs steps on childrens buffers

IF not possible (has lost leaf as on by child),

Add to list of leaves to be cleaned up. (removing child)

Else for each child:

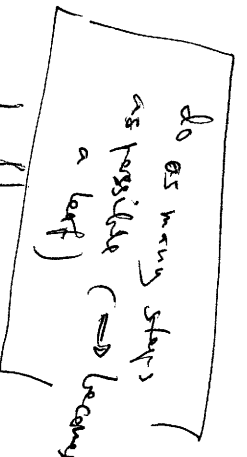
IF buffer size  $< n/2$

IF not leaf:

## Refill (n)

Else

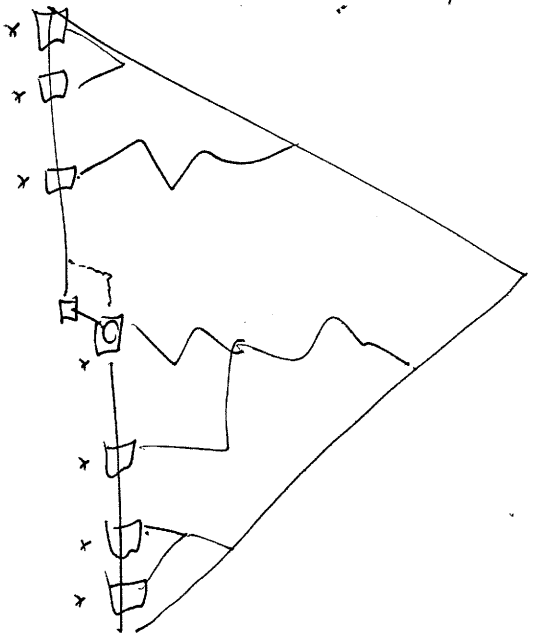
Add to list of leaves to be cleaned up.



[Remark: this is from ~~last~~ leaf

from disk steps leaves (again - size of childrens buffers)]

~~After~~ recursive  
return, before  
CleanUpLeaves :



$x =$  leaves to clean up. ( = all leaves with  $< \frac{n'}{2}$  desc. in subtree )

[ "max full" ]

CleanUpLeaves

current

While  $\exists$  unvisited leaf  $u \neq$  last leaf  $u$  :

If  $|u_1| + |u_2| > n'$  :  
{ merge change during }  
CleanUpLeaves

have  $n - |u_1|$  strongest desc in  
's subtree to u's subtree (merge)

SiftUp(u)

Else if  $\frac{n'}{2} < |u_1| + |u_2| \leq n'$  :

Merge u's subtree into u  
Delete(u)  
SiftUp(u)

Else : //  $|u_1| + |u_2| < \frac{n'}{2}$

Same as

# Analysis

$N$  elem. geht  $\rightarrow$  max.  $\frac{N}{n/2} + 1$  bucket

$\underbrace{\hspace{2cm}}$   
alle andere  
wert  $\geq n/2$   
elem. geht.  
 $\uparrow$   
sicht bucket

Heuride  $\leq \log_{\text{formant}} (\# \text{ bucket}) + O(1)$

$$= O\left(\log_{N/4} \left(\frac{N}{n}\right)\right) \cdot \left[ \left[ 1 + o(1) \right] \cdot \log_{N/4} \left(\frac{N}{n}\right) + O(1) \right]$$

falsch.



$O(n)$  inserts w/. next input buffer determining

Disse labels alt :

1)  $O(1)$  sift w/ps  $\rightarrow$  direkt after determining. Prio: heuride.  
 $\underbrace{\hspace{1cm}}$  Scan  $(n)$

~~the~~ linear w/ps  
of to sort w/ps  
selvance of heuride  
 $\leq n'$

2) Left of bucket (quad: tree)

under Refill : Da  $n/2$  merge steps bucket

forward + scan ( $\frac{n}{2}$ ) I/O set

$$\left[ \begin{array}{l} \text{less to look for} \\ \text{in next level} \\ \text{of leaves in next level} \end{array} \right] = O\left(\frac{n}{5}\right) + O\left(\frac{n}{5}\right) = O\left(\frac{n}{5}\right).$$

That leaves  $n/2 = O(n)$  number.

$\Rightarrow$  these number should be  $O\left(\frac{1}{5}\right)$  per

inversion, thus in all should be number

have  $O\left(\frac{1}{5} \log_{5/4} (N/n)\right)$  credits need to

be added in each iteration in level under turning of input leaflet.

NB: Under sift up + leaflet included in total in forward set. Leaf credits set in leaflet.

$\Rightarrow$  Credits leverage sig low spread.

3) Merge of Sift up and Clean Up leaves.

Benchmark of under Clean Up leaves has an number less more simple level of figure also.

(class. case w) to gauge I can will more to gauge

i checked, for Jan clothes.

Load settle like basket and at loose bundle  
not credits diff 2 x Sift UY (same  
dominate 2 x merge). Sled heaves  
used equivalent of rug bundle, under  
in put buffer forming.

Alt i alt re alt basket unit

due to  $\Theta\left(\frac{11}{B} \cdot \log_{n/B}\left(\frac{N}{B}\right)\right)$  credits

did nothing used as in put buffer forming.

DUS. have insert sled heave  
 $\Theta\left(\frac{1}{B} \cdot \log_{n/B}\left(\frac{N}{n}\right)\right)$  credits

