Soft Sequence Heaps

(Accepted to SIAM Symposium on Simplicity in Algorithms, SOSA21)

Gerth Stølting Brodal
Aarhus University
Heap

ExtractMin = lowest intersected line

Soft Heap

Soft heap properties

- ExtractMin can increase values (corruptions)
- Returns new corruptions
- \( \leq \varepsilon N \) corrupted elements in soft heap, \( 0 \leq \varepsilon \leq \frac{1}{2} \), \( N = \# \) insertions

\( \uparrow \) Insert\((x)\)

\( \downarrow \) ExtractMin()
# Soft heap results

<table>
<thead>
<tr>
<th>Soft heaps</th>
<th>INSERT / EXTRACTMIN</th>
<th>Reference</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduced car-pooling Binomial trees</td>
<td>O(log $\frac{1}{\varepsilon}$) / O(1)</td>
<td>Chazelle ESA98*/JACM00 *2018 ESA Test-of-Time award</td>
<td>Selection MST $O(m \cdot \alpha(m, n))$</td>
</tr>
<tr>
<td>&quot;A simpler ... soft heaps&quot;</td>
<td></td>
<td>Pettie, Ramachandran JACM02</td>
<td>MST optimal Unknown complexity</td>
</tr>
<tr>
<td>&quot;Soft heaps simplified&quot;</td>
<td></td>
<td>Kaplan, Zwick SODA09</td>
<td></td>
</tr>
<tr>
<td>Report corruptions</td>
<td></td>
<td>Kaplan, Kozma, Zamir, Zwick SOSA19</td>
<td>Heap selection (and related)</td>
</tr>
<tr>
<td>Soft sequence heaps</td>
<td></td>
<td>Brodal SODA21</td>
<td>Simplifying Frederickson JCSS93</td>
</tr>
</tbody>
</table>
Application of Soft Heaps – $O(n)$ Selection

```plaintext
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
        x = Q.EXTRACTMIN()
        small, large = partition(A, x)
    if k ≤ |small| then
        return select(small, k)
    return select(large, k - |small|)
```

Pivot $x$ has rank in $\left[\frac{|A|}{3}, \frac{2|A|}{3}\right]$

(x is the increased value)

$T(n) \leq T(2n/3) + O(n)$
Application of Soft Heaps – $O(k)$ Heap Selection

```plaintext
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 ∪ {e}
        for e ∈ C do
            Q.INSERT(e.left)
            Q.INSERT(e.right)
        S = S ∪ {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 $r \implies$ merge, rank $r+1$
- Rank $r$ list $\leq 2^r$ values
- $N$ INSERT $\implies$ rank $\leq \log N$

**INSERT**
Create rank 0 list containing $x$
while two list have equal rank do
merge the two lists

**EXTRACT_MIN**
Find list with smallest head element $e$
Remove and return $e
Soft Sequence Heap

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

**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 \frac{1}{\varepsilon} then
prune list

**EXTRACTMIN()**
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)
Suffix-min pointers and witness sets

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

Partial order of bounded “width”
Open Problems

- I/O & cache oblivious soft heaps with $O(B)$ soft heap operations take $O(1)$ I/Os?

- Other applications of soft heaps?

- Are Soft Heaps simpler?