1 Funnel Heap

- Introduced by Brodal et al. [1].
- Based on merging instead of distribution.

References

2 Funnel
3 Structure

\[(k_1, s_1) = (2, 8), \quad (1)\]
\[s_i = s_{i-1}(k_i + 1), \quad \text{and} \quad (2)\]
\[k_i = \lceil s_i^{1/3} \rceil, \quad (3)\]

where \(\lceil x \rceil\) is “\(x\) rounded to the nearest power of 2”.
4 Extract

- Fill $A_1$ if it is empty.
- Extract from $A_1$

5 Insert

- Insert element in $I$.
- If $I$ is full perform sweep.
6 Sweep

Idea: Merge elements, and move them to a higher link.

1. Find first empty S buffer.
2. Create $\sigma_1$ and $\sigma_2$, and merge them to form $\sigma$.
3. Insert elements on path from the root to the empty S buffer.
7 Space Complexity

- In a link $i$ there are $k_i$ S buffers of size $s_i$.
- By the definition of $k$ and $s$ we know that: $k_i = O(s_i^{1/3})$.
- The space consumption of the S buffers are thus $O(k_i^4) = O(s_i^{4/3})$.
- Since $K_i$, $A_i$ and $B_i$ are all of size $O(s_i)$, the space in the S buffers dominates the space used for a link $i$.
- Since $s_i$ and $k_i$ is increasing, the space consumption of a Funnel Heap with $i$ links is dominated by the $i$th link.
- Since the space consumption of a Funnel Heap with $i$ links is $O(k_i^4)$, and $k_{i+1} = O(k_i^{4/3})$, the worst case space complexity is $O(N^{4/3})$. 
8 I/O Complexity

They prove that the amortized I/O complexity of an extract is

$$O \left( \frac{1}{B} \log_{M/B} \left( \frac{N}{B} \right) \right).$$

Which is optimal for cache-oblivious priority queues.