Number systems and de-amortization [Knuth 1970]

Use quaternary instead of a binary counter

<table>
<thead>
<tr>
<th>b_5</th>
<th>b_4</th>
<th>b_3</th>
<th>b_2</th>
<th>b_1</th>
<th>b_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Increment operation

Goal: Make increment constant time
Problem: 00 11111111
        01 00000000

Example application to data structure

Binomial tree of rank k

\[ B_0 \ (rank \ 0) : 0 \]
\[ B_1 : 0 \]
\[ B_2 : 0, 0 \]
\[ \ldots \]
\[ B_k : B_{k-1} B_{k-2} B_{k-3} \ldots B_0 \]

Binomial heap

Collection of binomial trees, each tree has unique rank

e.g.: \[ B_5 B_3 B_2 B_1 B_0 \]

- Each node stores a key
- Trees are in heap order

\[ |B_k| = 2^k \]

\[ |B_k| = 1 + \sum_{i=0}^{k-1} |B_i| \]

Binomial heap

Collection of binomial trees, each tree has unique rank

e.g.: \[ B_5 B_3 B_2 B_1 B_0 \]

- Each node stores a key
- Trees are in heap order

\[ |B_k| = \binom{k+1}{2} \]

\[ \text{View as representation of } n \text{ in binary} \]

\[ 5 + 3 + 2 + 1 = (10111)_2 \]
Insert*(k) - Create a new tree B₀
- Add the tree to collection of trees
- Combine trees of equal rank until done

Amortized cost of Insert:
\[ O(1) \]

[Would like actual cost of the time]

Merge (combine trees of equal rank) until done

<table>
<thead>
<tr>
<th>Time</th>
<th>[ O(k) = O(\log n) ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[</td>
</tr>
<tr>
<td></td>
<td>[ k = O(\log n) ]</td>
</tr>
<tr>
<td></td>
<td>[ k \text{ here is rank of largest tree} ]</td>
</tr>
</tbody>
</table>
- Cut out root with min key
- Merge children-first with rest of heap

Find-min \( O(1) \)
Insert \( O(1) \) amortized
Merge \( O(1) \)
Extract-min \( O(1) \) amortized

\[ \text{de-amortize?} \]
\[ \text{actual } O(1) \text{ true} \]

Binary counter

\[ n = \text{value} = \sum_{i=0}^{k} b_i \cdot 2^i \leq 2^{k+1} - 1 \]

\[ t_k \ldots t_2 t_1 t_0 \]

\[ m = \sum_{i=0}^{k} t_i \cdot 2^i \leq \sum_{i=0}^{k+1} 2^{i+1} = 2^{k+2} - 1 \]

Idea: Use digits \( \{0, 1, 2\} \)

\[ 01111111 \]
\[ 01111112 \]
\[ 011111124 \]

\[ 02222222 \]

[Still had to flip many digits]

Idea: Keep 2s separated (cannot be neighbors)
Rules for number system

(R1) Digits \{0, 1, 2\}

(R2) Between every two 2's, there is at least one 0
    (and arbitrary # of 1's)

(R3) \( x_0 \neq 2 \) (last digit not 2)
    e.g. \( x_0 \) \( x_1 \) \( x_2 \) \( x_3 \) \( x_4 \) \( x_5 \) \( x_6 \) \( x_7 \) \( x_8 \) \( x_9 \) \( x_{10} \)
    2 0 1 0 1 2 0

How to increment counter?

increment
\[ x_0 \rightarrow x_0 + 1 \]
if \( x_0 = 2 \)
\[ x_0 = 0 \]
Fix

Fix

Store indices of all 2-digits

if \( |S| = 0 \)
\[ x_1 \rightarrow x_1 + 1 \]
if \( x_1 = 2 \)
push \( (1, S) \)
else
if \( x_1 = 0 \)
\[ x_1 = -1 \]
elif \( x_1 = 1 \)
\[ x_1 = -2 \]
last \( \rightarrow \) pop \( (S) \)
last \( \rightarrow 0 \)
last+1 \( \rightarrow \) last+1 + 1
if \( \text{last+1} = 2 \)
push \( (1, S) \)

Example of number system & redundant

\[ x_2 \rightarrow x_0 \]
\[ x_1 \rightarrow x_1 \]
\[ x_0 = 0.2^0 + 2.2^1 + 1.2^2 \]
\[ = 8 \]

\[ \cdots 2 0 1 0 1 2 1 \]
\[ \cdots \]

\[ \left[ \begin{array}{c} 1 \\ \vdots \\ S \end{array} \right] \]

Stack S

- Value = Value + 1 + 2 = Value + 1

\[ \begin{array}{c} 1 \\ \vdots \\ S \end{array} \]

- Value unwrapped

\[ x \in \{ 0, 1 \} \]
\[ R_1 \leq x \leq R_3 \]
if \( k_1 = 2 \), \( x_i \) will be popped
\[ \text{pop}(S) \rightarrow \]
\[ t_1 \rightarrow 1 \]
\[ t_2 \rightarrow t_2 + 1 \]
\[ \text{if } (t_2 = 2) \text{ push } (2, 5) \]
\[ t_5 \rightarrow t_5 \]
\[ t_6 \rightarrow t_6 \]
\[ \text{for } \frac{3}{2} \]
\[ \text{by } 3 \]
\[ \text{decrease by } 3 \]
\[ + 5 \]
\[ + 1 \]
\[ R_1 \subseteq \]
\[ x = 0 \]
\[ R_2 \subseteq \]
\[ x = 1 \]
\[ x = 0 \]
\[ x = 1 \]
\[ 2^{x+1} - 2 \cdot 2^k = 0 \]

Back to binomial heaps

2.9.
\[ B_5, B_3, B_2, B_0 \]

2.9.
\[ B_5, B_3, B_2, B_1 \]

New version:
Each real opers at most \( \text{x times} \) twice

- Rules 1-3 have natural interpretation
- Data structure insert can follow increment/fix from before:
\[ t_1 \rightarrow t_2 \]
\[ x \cdot 2 \]
\[ (x) \circ \]

Result:
0(1) reached time insert
0(\log n) time extract-min? merge?

We need to add two counters, such that Rules 1-3 are true for the result:

exercise: Show how to do add

\( \rightarrow O(1/2) \text{ Number system vs binomial heap) } \]

a showcase of de-amortization.
Technique/idea more generally applicable.