Triggering a cascading cut of length $k = 6$ in a Fibonacci heap
Step 1 - Add nodes

First, we will add enough items to allow for a path from root to leaf with \( k = 6 \) intermediate nodes to exist. This means \( 2^{k+1} = 2^7 = 128 \) nodes. We will add one more than that in order to have only a single (large) tree after Step 2.
Step 1 - Add nodes - $\text{INSERT}(0)$
Step 1 - Add nodes - $\text{INSERT}(1)$
Step 1 - Add nodes - 126 INSERTs later
Step 1 - Add nodes - \textsc{Insert(128)}
Step 2 - Consolidation

We now perform an $\text{EXTRACT-MIN}$ operation to force a consolidation step which will merge our singleton trees into a single tree with depth $k + 1 = 7$. 
Step 2 - Consolidation - Result
Step 3 - Strategic cuts

We will perform $k - 1 = 5$ \texttt{Decrease-Key} operations on $k - 1 = 5$ carefully chosen nodes. These nodes will be cut from the start tree and appended to the root list. As a result of this, certain nodes in the start tree will be marked. This prepares the cascading cut which we will trigger in the next step.
Step 3 - Strategic cuts - Before first cut
Step 3 - Strategic cuts - After $\text{DECREASE\text{-}KEY}(126,-126)$
Step 3 - Strategic cuts - After \textbf{DECREASE-KEY}(123,-123)
Step 3 - Strategic cuts - After `DECREASE-KEY(117,-117)`
Step 3 - Strategic cuts - After \texttt{DECREASE-KEY}(105,-105)
Step 3 - Strategic cuts - After \textbf{DECREASE-KEY}(81,-81)
Step 4 - Triggering the cascading cut

Lastly we need to trigger our cascading cut. This is done by removing (ie. Decrease-Key so that it violates the heap property) a child (in this case item 127) of the bottommost node in the chain of marked nodes we have set up in Step 3.
Step 4 - Triggering the cascading cut - Before
Step 4 - Triggering the cascading cut - After