$O(\frac{\lg w}{\lg\lg w})$ data structure [Beame & Fich – JCSS 2002] (matching their $\forall w \exists n$ LB)

6.897
L12.2
Mar. 15, 2005

- break query word into k equal-size chunks
  ↳ to be chosen later
- send $\Theta(a/k)$-bit hash of each chunk $\Rightarrow \Theta(a)$ bits ✓
- Bob can detect w.h.p. whether query matches any of $2^{\Theta(a/k)}$ most common values for each chunk
  ↳ heaviest edges in height-k tree

$\Rightarrow$ find first chunk where query falls of heavy paths via one probe to Bob, w.h.p.

- catch: w.h.p. in terms of $2^{\Theta(a/k)}$, not n
- convert into w.h.p. in terms of k:

$$2^{\Theta(a/k)} > k^c \iff \Theta(a/k) > c \lg k \quad (a \geq \lg n)$$
$$\impliedby \Theta(\lg n / k) > c \lg k$$
$$\iff k \lg k < \Theta(\lg n)$$
$$\iff k < \Theta\left(\frac{\lg n}{\lg\lg n}\right)$$

- check claimed match on heavy paths with lcp
- on failure (prob. $O(1/k)$), walk down "by hand" $\rightsquigarrow O(k)$ time $\Rightarrow O(1)$ expected
- second probe to find whether mismatch in heavy paths from match with light edge $\Rightarrow$ search in light subtree, $n' = n/2^{\Theta(a/k)}$
  or from no match at all $\Rightarrow$ search in chunk, $w' = w/k$
- cost $= O(\max\{\lg_k w, \lg_{2^{\Theta(a/k)}} n\})$
- set $\frac{\lg w}{\lg k} = \frac{\lg n}{\Theta(a/k)} = \Theta(k)$ by setting $a = \Theta(\lg n)$ (poly. space)

$$\Rightarrow k \lg k = \Theta(\lg w)$$
$$\Rightarrow k = \Theta\left(\frac{\lg w}{\lg\lg w}\right)$$
$$\Rightarrow \lg_k w = \Theta\left(\frac{\lg w}{\lg\lg w}\right)$$

- space: $2^{\Theta(a)} = n^{\varepsilon}$ for lookup table for first probe (for suitable consts.)
  $\times$ $n \cdot t$ different subsets/wordsizes considered (as before)
  $= O(n^{1+\varepsilon'})$ for $n \geq w^{\Theta(1)}$ ↝ necc. for w.h.p. assumption
  (else $O(1)$ via fusion trees)