Approximate string matching [Cole, Gottlieb, Lewenstein — STOC 2004]

6.897
L 19.1
Apr. 14, 2005

- given error tolerance k
- preprocess text T
- query: find occurrences of P in T within error k.
  - Hamming distance: # mismatches
  - edit distance: # insertions/deletions/substitutions
- solution: $O(|T|\,(c \lg |T|)^k / k!)$ space & preprocessing
  $O(|P| + (c \lg |T|)^k \lg\lg |T| + 3^k (\#\text{occurences}))$ query
  ↑ only for edit distance

FOCUS HERE ✱ - subproblem of independent interest:
  - P has up to k "don't care" characters (? wildcards)
  - want "exact matches" (? matches anything)
  - solution: $O(|T| \lg^k |T|)$ space
    $O(|T| \lg^k |T| + |T| \lg |\Sigma|)$ preprocessing
    $O(2^k \lg\lg |T| + |P| + \#\text{occurrences}))$ query

Simple solution: $|\Sigma|^k \cdot |P|$ query
- walk suffix tree of T as usual
- upon ? in P at branching node of suffix tree, recurse into all $\le |\Sigma|$ subtrees

Better solution: $2^k \cdot |P|$ query
- heavy-light decomposition on suffix tree of T (primary tree)
- each node in primary tree stores secondary tree on union of light subtrees hanging off node, excluding first character
- recurse k times → $k+1$ levels of trees
- light depth $\le \lg |T|$
  ⇒ each leaf appears in $\le \lg^k |T|$ trees
  ⇒ $O(|T| \lg^k |T|)$ space (& preprocessing)
- search now branches two ways upon ? in P:
  ① heavy edge
  ② secondary tree representing all light edges

$2^k \lg\lg |T| + |P|$ solution uses "level ancestor" data structure (in uncompressed trie)