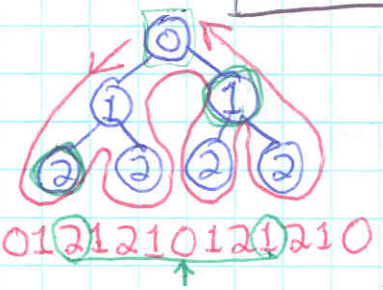
6.897
L15.2
Mar. 31, 2005

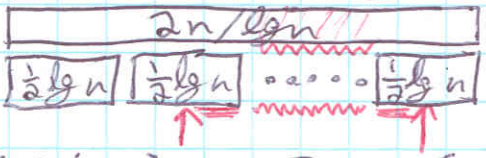
LCA [Bender & Farach-Colton — LATIN 2000]

① Reduce to RMQ, "±1 case"
- take Euler tour of tree
- store depths of nodes as visited
- LCA(i,j) = (index of) min depth in range i ... j



Extra property: adjacent values (depths) differ by ±1.

② Indirection: Split into groups of $\frac{1}{2}\lg n$



← another RMQ DS on min's of groups

- RMQ(i,j) = min { RMQ(i, ∞) in i's group,
  RMQ(−∞, j) in j's group,
  RMQ(i's group, j's group) in top }
  — [2i/lg n]

③ Lookup table for groups
- index of RMQ(i,j) invariant under translation: ±k on each value
  ⇒ translate so that first number is 0
  ⇒ $2^{\frac{1}{2}\lg n} = \sqrt{n}$ distinct group types (±1's)
- $(\frac{1}{2}\lg n)^2$ queries, output of $O(\lg\lg n)$ bits
  ⇒ lookup table (group type, i, j) ↦ RMQ index
  uses $O(\sqrt{n}\lg^2 n \lg\lg n)$ bits $= o(n)$ bits.

④ Trivial RMQ: store all answers $(\forall i,j)$
- $O(1)$ time
- $O(n^2)$ space

⑤ Smarter RMQ: $O(n\lg n)$ space, $O(1)$ time [top of indirection]
- store answer from any start point ← n choices
  of interval of length = power of 2 ← lg n choices
  ⇒ $O(n\lg n)$ space
- any interval of length k can be covered by two intervals of length $\lfloor\lfloor k\rfloor\rfloor = 2^{\lfloor\lg k\rfloor}$