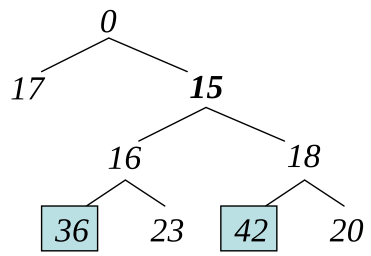
Lecture 16: static RMQ, LCA

Range Minimum Queries (RMQ):

- static array A of n numbers
- query $\text{RMQ}(i, j) = \min\{A[i], \ldots, A[j]\}$
- naïve solution: $O(n^2)$ lookup table, $O(1)$ query time
- cartesian tree: [Gabow, Bentley, Tarjan - STOC 84]

- root = min in array, say $A[i]$
- left subtree: recurse on $A[1], \ldots, A[i-1]$
- right subtree: recurse on $A[i+1], \ldots, A[n]$
- $O(n)$ time construction (next lecture)

A = [*17, 0, 36, 16, 23, 15, 42, 18, 20*]



$\Rightarrow$ $\text{RMQ}(i, j)$ reduces to $\text{LCA}(i, j)$

Lowest Common Ancestor (LCA):

- static tree on n nodes
- $O(1)$ time/query, $O(n)$ space [Harel and Tarjan - SICOMP 84, Berkman and Vishkin - SIAM J. Comput 1993]
- simplification [Bender and Farach-Colton - LATIN 00] (today)
- dynamic: $O(1)$ insert/delete leaves, $O(1)$ subdivide/merge edges [Cole and Hariharan - SODA 99]

LCA in a complete BST

- word RAM model
- $\text{LCA}(a, b)$ found by most significant bit of $a \oplus b$

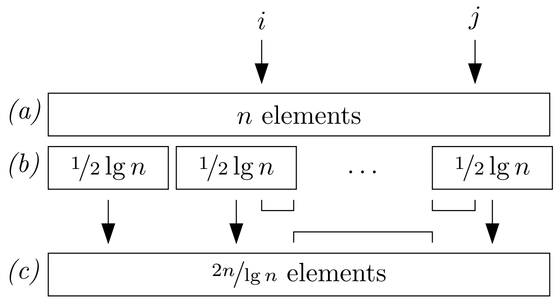
Reduce LCA to RMQ±1

- RMQ±1: adjacent elements differ by ±1
- take Euler tour of the tree and store depths of visited nodes.

0	17	0	15	16	36	16	23	16	15	18	42	18	20	18	15	0
0	1	0	1	2	3	2	3	2	1	2	3	2	3	2	1	0
									⇑							

- LCA(i, j) = index of min depth in range

Solving RMQ±1 via Indirection



- split into groups of $\frac{1}{2} \lg n$
- a summary RMQ DS on the $2n/\lg n$ minimums

$$RMQ(i, j) = \min \begin{cases} RMQ(i, \infty) \text{ in } i\text{'s group} \\ RMQ(-\infty, j) \text{ in } j\text{'s group} \\ RMQ(> i\text{'s group}, < j\text{'s group}) \text{ in the summary structure} \end{cases}$$

Lookup Table for Groups

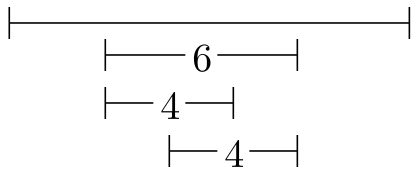
- index of RMQ(i, j) is invariant under translation
  $\Rightarrow$ subtract $A[1]$ from all $A[i]$'s so $A[1] = 0$
  $\Rightarrow 2^{\frac{1}{2}\lg n} = \sqrt{n}$ distinct group types (±1)
- $1 \le i, j \le \frac{1}{2} \lg n$ so $(\frac{1}{2} \lg n)^2$ possible group queries
- group query output requires $O(\lg \lg n)$ bits
  $\Rightarrow$ lookup table requires $O(\sqrt{n} \lg^2 n \lg \lg n) = o(n)$ bits.

RMQ on the Summary Structure

- adjacent elements in the summary are not necessarily ±1
- trivial solution of storing all answers $\forall(i, j)$ is not good enough $(O(n^2/\lg^2 n))$
- we need general RMQ DS of $O(n \lg n)$ space & time and $O(1)$ query $(|summary| = 2n/\lg n)$

- store answer from any start point (n choices) and interval of length = power of 2 ($\lg n$ choices) $\Rightarrow O(n \lg n)$ space (and time via dynamic programming)

- any interval of length k can be covered by two (possibly overlapping) intervals of length $2^{\lfloor \lg k \rfloor}$



Before and after [1]

Before: [Berkman and Vishkin - SIAM J. Comput. 1993]
- split into groups of $\lg n \Rightarrow$ lookup table is too big
- split every group into groups of $\lg \lg n$

After: [Fischer and Heun - CPM 2006]
- avoid the cartesian tree needed for $RMQ \rightarrow LCA \rightarrow RMQ \pm 1$
- groups are no longer in a ± 1 form
- construct all possible cartesian trees of $s = \frac{1}{4} \lg n$ elements (Catalan number $\leq 4^s$).

References

[1] M. A. Bender, M. Farach-Colton, *The LCA Problem Revisited*, LATIN 2000: 88-94.

[2] O. Berkman and U. Vishkin. *Recursive star-tree parallel data structure*, SIAM J. Comput., 22(2), 1993.

[3] R. Cole, R. Hariharan, *Dynamic LCA Queries on Trees*, SODA 1999: 235-244.

[4] J. Fischer, V. Heun, *Theoretical and Practical Improvements on the RMQ-Problem, with Applications to LCA and LCE*, CPM 2006: 36-48.

[5] H. N. Gabow, J. L. Bentley, R. E. Tarjan, *Scaling and Related Techniques for Geometry Problems*, STOC 1984: 135-143.

[6] D. Harel, R. E. Tarjan, *Fast Algorithms for Finding Nearest Common Ancestors*, SIAM Journal on Computing, 13(2): 338-355, 1984.