Review for Test 1.

10.16

Range Minima Queries

- Segment Trees
  - Subtree for first half on left, second half on Right
    - recursively
    - Know how to construct by hand
    - Know what nodes you need to examine (path) by hand - keep them minimal
      $O(\lg n)$ query

- Sparse Trees ("2^i range precompute")
  - know how to construct a table
  - Know how to do a query
  - why does it take $O(1)$ time to query?
  - How much space... if indices are $\lg n$ in size?

- Array Blocking $s = \frac{\lg n}{4}$

What is the Look-up Table?
How big is it?
What does a query constitute?
How did we get $O(\lg n)$, when $\exists$ $O\left(\frac{n}{\lg n}\right)$ entries in Look-up table?

How do we get linear space?
linear time precompute?

- Cartesian Trees
  - Know how to construct one out of a given array A.
  - Know how to search one for a min in a range

Lowest Common Ancestor LCA
- RMQ $\leq_{\substack{O(1) \\ \text{query}}}$ LCA — show?
- LCA $\leq_{\substack{O(1) \\ \text{query}}}$ RMQ±1

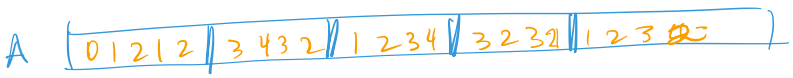
Claim: The RMQ±1 on Eulerian tree depth array is LCA of tree.

Proof: For you to do.

Given an array
- construct Cartesian Tree
- Do Eulerian tour, construct ±1 array of node depths.

RMQ strategy involves blocks of size $\frac{\lg n}{2}$

Summary Array



Min Contig Sum

$-[0\ -1\ -2\ 2\ -5\ 4\ 11\ -6\ \cdots$

Amortized Analysis

multipop stack.

- accounting method
- Φ potential method.

Binary Counter

Binary Representation Heap

Union Find

- what is amortized running time for n ops? ←
- might do a simplified "tower of 2's" analysis of some DS. - know $\lg^* n$.
- Know path compression, union-by-rank, & proof that union-by-rank guarantees that $size(x) \geq 2^{rank(x)}$

also # nodes of rank r is $\leq \frac{n}{2^r}$.

Fib heaps

Know what the DS looks like after a series of ops from {Insert, ExtractMin}

Ignore marks for now.