Lowest common ancestors
Write an Euler tour of the tree

LCA(1,5) = 3

Shallowest node
Range minimum

Preprocess an array, such that given $i,j$ you can find the minimum in $[i,j]$ fast.

Reduction takes linear time.
Trivial algorithms for RMQ
Less trivial algorithms to RMQ

• Try to use $O(n \log(n))$ space to do a query in $O(1)$ time
Optimal solution

\[ \frac{2n}{\log n} \] blocks

Remember the min in each block (A') and its position (B)

B[0]  ...  B[i]  ...  B[2n/logn]

A'[0]  ...  A'[i]  ...  A'[2n/logn]
Example

\[ n = 16 \]

\[ A[] : \begin{bmatrix} 10 & 25 & 22 & 7 & 34 & 9 & 2 & 12 & 26 & 33 & 24 & 43 & 5 & 11 & 19 & 27 \end{bmatrix} \]

\[ \frac{2n}{\log n} \text{ blocks} = 8 \]

\[ A'[] : \begin{bmatrix} 10 & 7 & 9 & \cdots \end{bmatrix} \]

\[ B[] : \begin{bmatrix} 0 & 3 & 5 & \cdots \end{bmatrix} \]
Preprocess $A'$ for RMQ using the $O(n \log(n))$ space algorithm.

Since the size of $A'$ is $\frac{2n}{\log n}$, this would take

$$\frac{2n}{\log n} \log(\frac{2n}{\log n}) = O(n)$$

Space, preprocessing time
How do we answer queries?

i and j might be in the same block, we need some mechanism to answer inside blocks

i < j on different blocks, answer the query as follows:

1. Compute minima from i to end of its block.
2. Compute minima of all blocks in between i’s and j’s blocks.
3. Compute minima from the beginning of j’s block to j.

Return the index of the minimum of these 3 values.
So what do we do inside blocks?
We need to solve a special case of RMQ

$$\pm 1$$ restriction
Each subproblem can be described by the first entry and a vector of $\pm 1$

\begin{array}{cccccccc}
0 & 1 & 2 & 3 & 2 & 1 & 2 & 3 & 2 & 1 \\
\hline
+1 & +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\
\hline
3 & 4 & 5 & 6 & 5 & 4 & 5 & 6 & 5 & 4 \\
\end{array}

Two subproblems with the same vector of $\pm 1$ are equivalent.
The bottom line

There aren’t too many different subproblems, only

\[ 2 \left( \frac{\log n}{2} - 1 \right) = O(\sqrt{n}) \]

For each such subproblem prepare all answers in advance \( \Rightarrow \)

\[ O\left( \sqrt{n \log^2 n} \right) = o(n) \]
Pick a subproblem:

\[ \sqrt{n} \]

Find the solution using \( i, j \):
Summary

Each block knows its subproblem ($A''$)

$\frac{2n}{\log n}$ blocks

$B[0] ... B[i] B[2n/\log n]$

$A'[0] ... A'[i] A'[2n/\log n]$

$A''[0] ... A''[i] A''[2n/\log n]$