Euler tour

- Eluer tour of tree encounters the node in following order:

  `a b e k e l e b f b g b a c h m h n`

  `h o h c a d i d j p j q j r j s j d a`.

- A node is visited once more than the number of children it has.

- If we list out the first instance of visit of each node in order then we get preorder traversal. For example, preorder list for above tree is `a b e k l f g c h m n o d i j p q r s`

- Listing out last instance of visiting nodes, we get postorder list. For example `k l e f g b m n o h c i p q r s j d a` is the postorder list.

- List out all but first and last instances of visiting nodes to get inorder list.

Euler tour

- We can create a linked list of Euler tour order for an ordered tree.
- Each tree edge can be replaced by a pair of directed anti-parallel edge.
- Assume that we can travarse an edge only in its direction of orientation.
- For generating the Euler tour start from root.
- At the current node choose the leftmost untraversed directed edge going down.
- If there is no remaining untraversed edge going down at the current node choose the reverse of the edge by which the current node is reached and traverse one level up.
- Repeat the two previous steps till the root is reached and no edge is left untraversed.

Euler tour

- While the abstract process of generating Euler tour seem precise, a suitable represent the tree is very essential.
- The representation should be able to store the information about reversal of an edge and the edges going out from each node.
- We take tree represented by circular linked lists of children for each node.
- Note that each member of an adjacency list of a node in the tree implicitly represents one of the two anti-parallel directed edge.
- Each member in the list also has a pointer that leads to the reversal of the edge represented by the node.