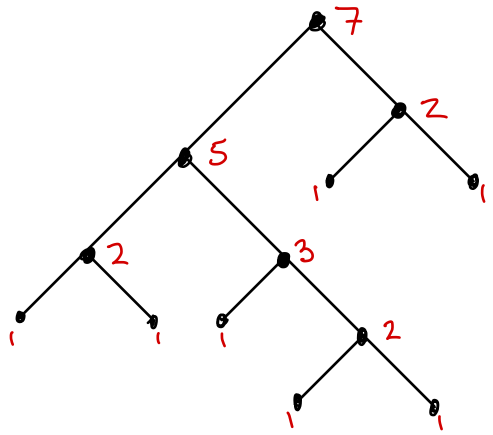
Indexing paths in a binary tree:

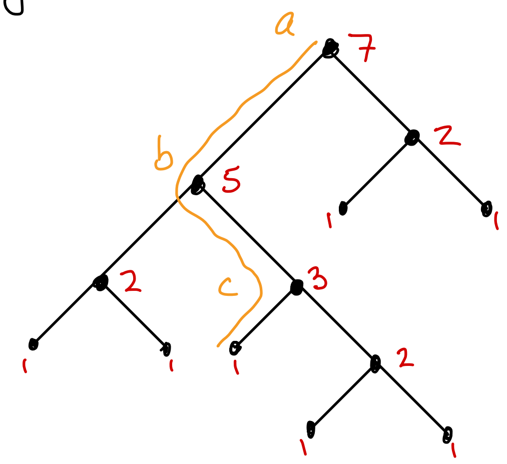
(ignoring the fact that to choose a path is to choose a leaf node)

Number of sub-paths below each node



(so there are paths w/ indices 1 through 7.

e.g. find path with index 3.



at (a) we go left because $5 \geq 3$

at (b) we go right because $2 < 3$, and we adjust the sub-index to $3-2=1$.

at (c) we go left because $1 \leq 1$

Algorithm: Indexing paths in a binary tree:

```
Inputs: A binary tree and an index
  Set the subindex i to the index
  Set current node to root node
  while current node is not a leaf node;
        let n be subpaths below left child of current node.
        If n ≥ i;
              set new current node to left child of old current node
        Otherwise;
              set new current node to right child of old current node
              set new subindex i to i - n.
The final 'current node' is the node with the provided index.
```