Proof of claim:

- left subtree has $\ell/2$ updates & $\ell/2$ random $\sqrt{n}$ permutations
- any encoding of these permutations must use $\Omega(\ell\sqrt{n}\lg n)$ bits
  [entropy / Kolmogorov complexity argument]
- we'll design an encoding strategy that is smaller than this if claim doesn't hold
- set up: - know the past (before v's subtree)
- note: if we know correct sums from right subtree, we can determine perms. from left subtree
- want to encode sums in right subtree.

Easier scenario: have operation sum(i) $\rightarrow \sum_{j=1}^{i} \pi_i$

- let $R = \{$cells read during right subtree$\}$
  $W = \{$cells written during left subtree$\}$
- encode $R \cap W$ → address & contents of each cell
- decoding strategy for sums in right subtree:
  - run sum operations (simulate)
  - to read cell written in right subtree → easy
    in left subtree → in $R \cap W$
    in past → known
- encoding requires $|R \cap W| \cdot O(\lg n)$ bits

$\Rightarrow |R \cap W| \cdot O(\lg n) = \Omega(\ell\sqrt{n}\lg n)$

$\Rightarrow |R \cap W| = \Omega(\ell\sqrt{n})$ ✓