09-XB

BST Application

Interval Tree

Your instinct, rather than precision stabbing, is more about just random bludgeoning.

邓俊辉
deng@tsinghua.edu.cn

Stabbing Query

- Given a set of intervals in general position

  on the x-axis: $S = \{ s_i = [x_i, x_i'] \mid 1 \le i \le n \}$

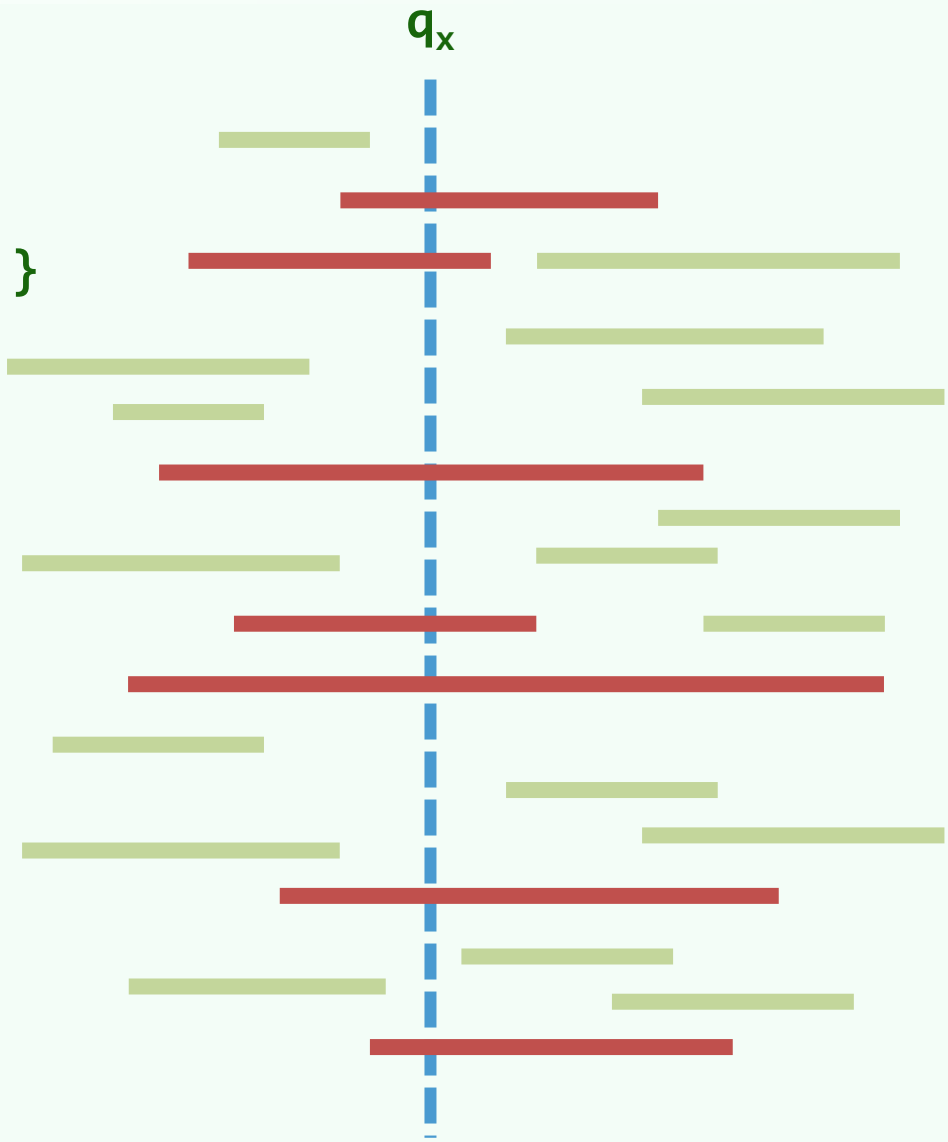
  and a query point q_x

- Find all intervals that contain q_x

  $\{ s_i = [x_i, x_i'] \mid x_i \le q_x \le x_i' \}$

- To solve this query,

  we will use the so-called interval tree ...

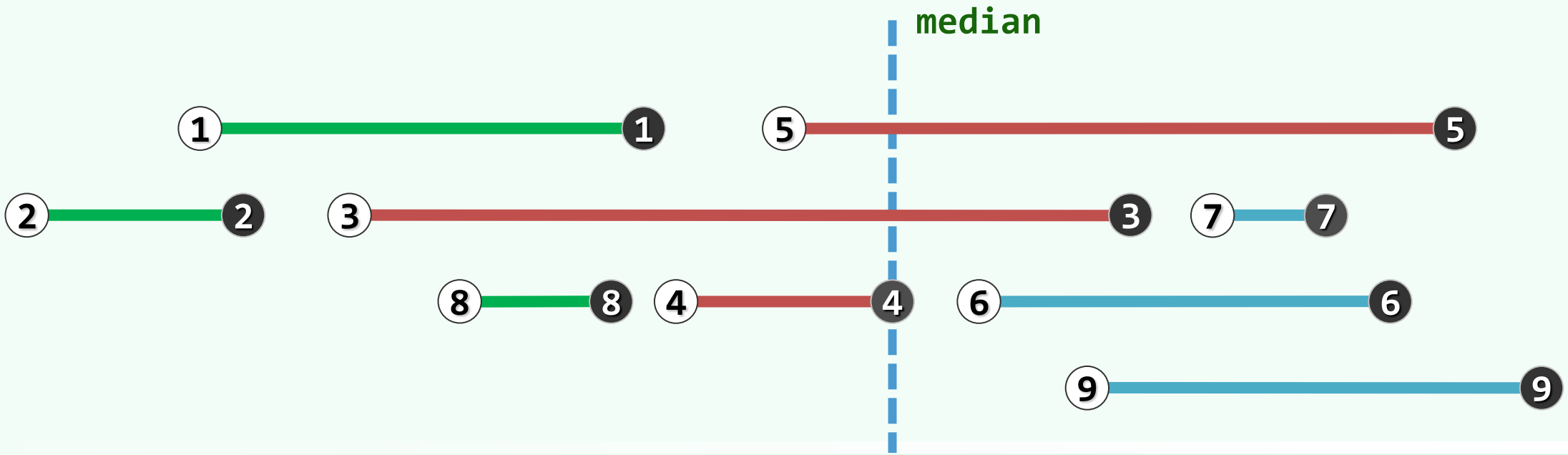


Median

- Let $S = \{ s_1, \ldots, s_n \}$ be the set of intervals
- Let $P = \partial S$ be the set of all endpoints

  // by general position assumption, $|P| = 2n$

- Let $median(P) = x_{mid}$ be the median of P

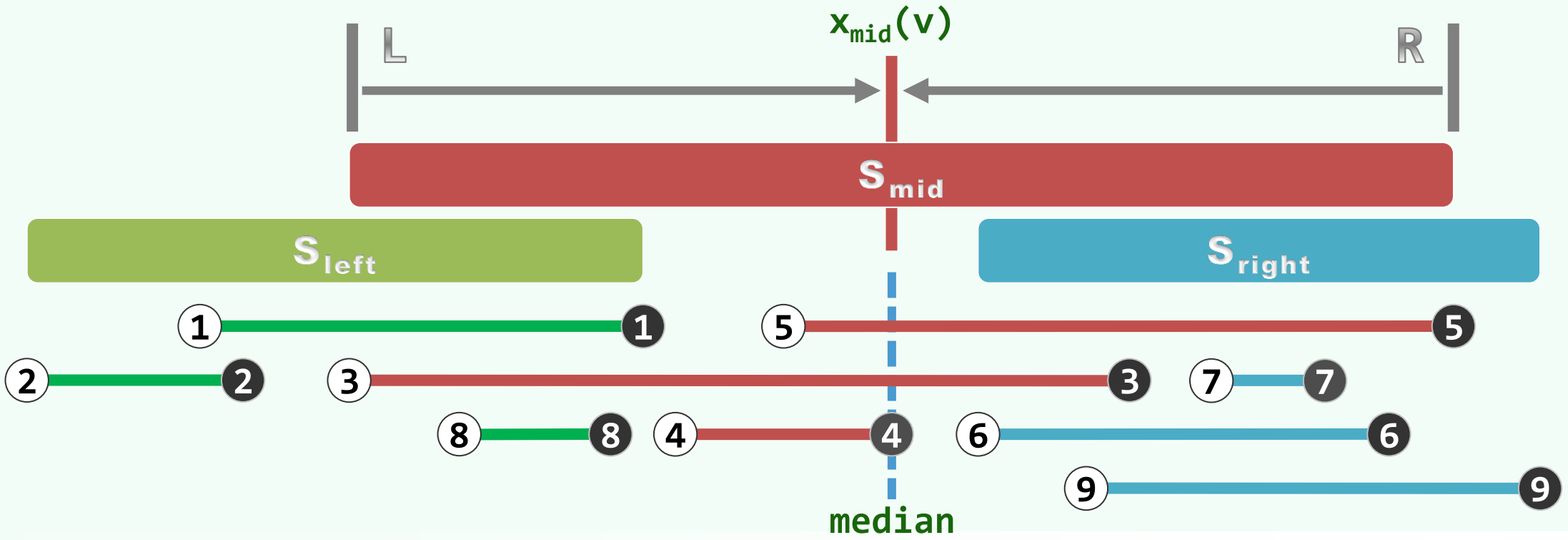


Partitioning

- All intervals can be then categorized into 3 subsets :

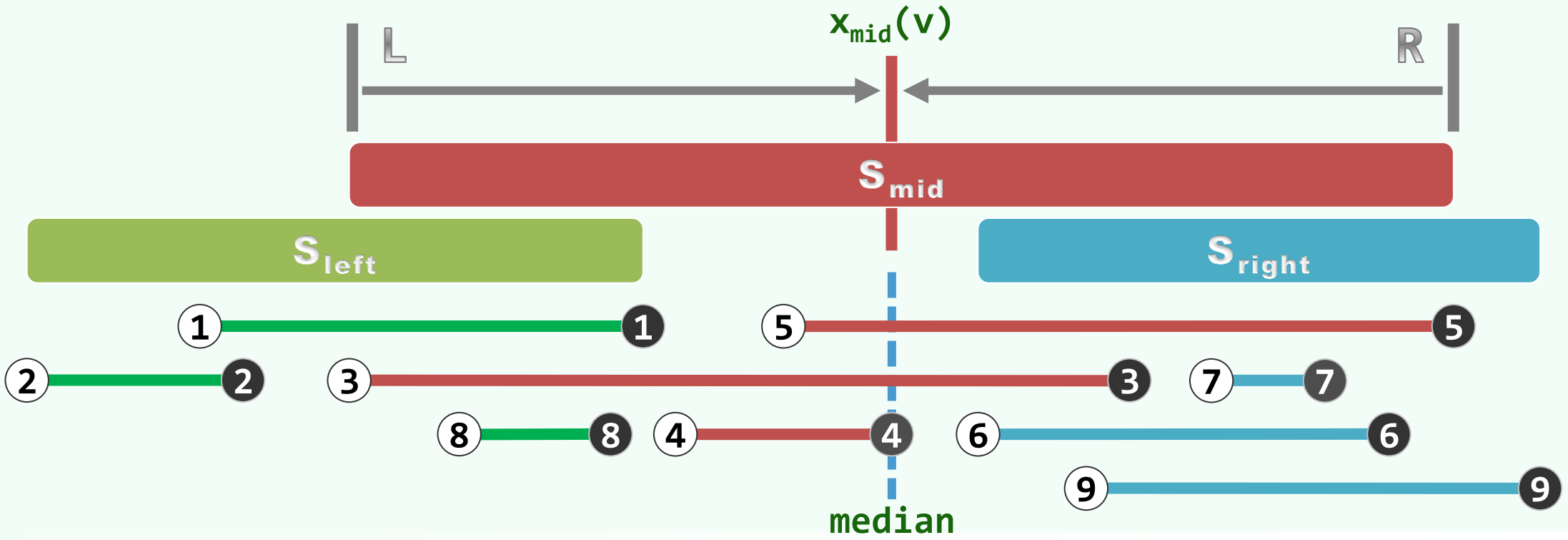
$$S_{left} = \{ S_i \mid x_i' < x_{mid} \} \quad S_{mid} = \{ S_i \mid x_i \le x_{mid} \le x_i' \} \quad S_{right} = \{ S_i \mid x_{mid} < x_i \}$$

- $S_{left/right}$ will be **recursively** partitioned until they are empty (leaves)



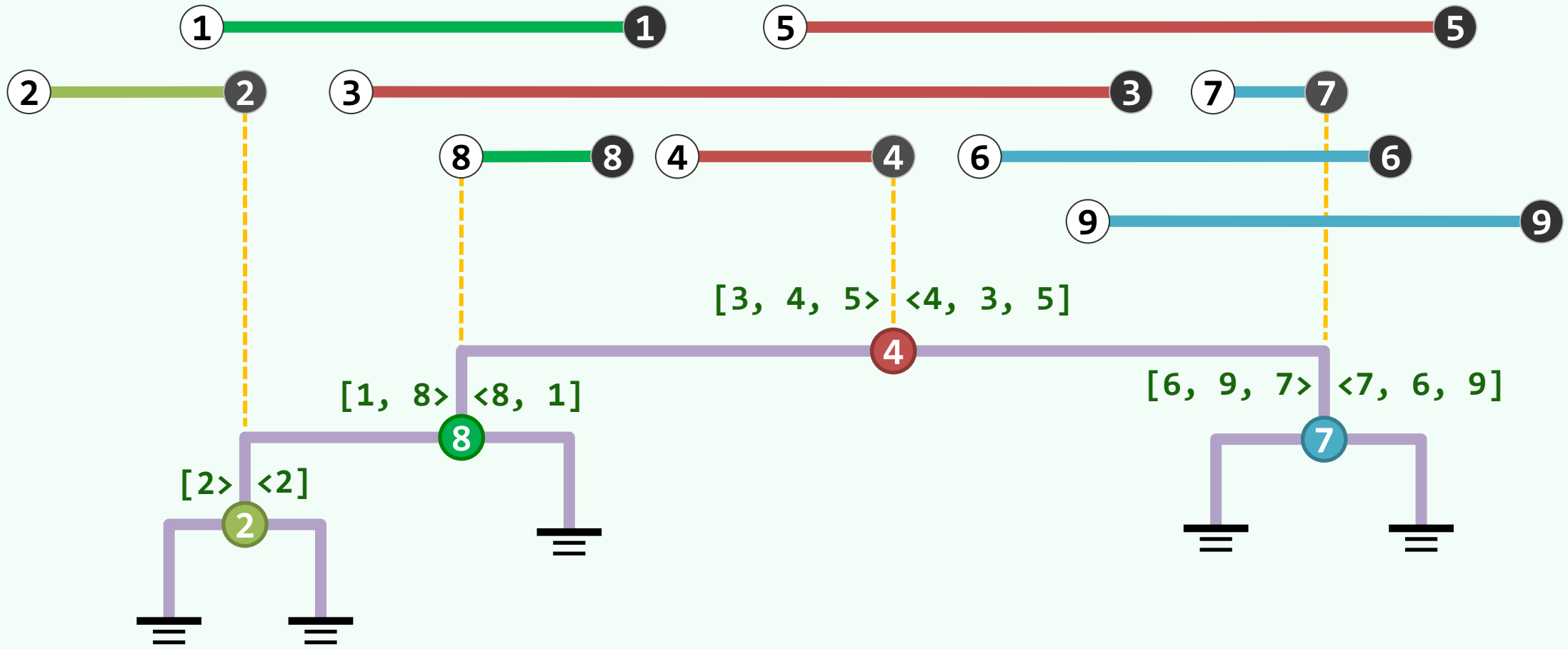
Balance

- $\max\{ |S_{left}|, |S_{right}| \} \leq n/2$
- **Best case:** $|S_{mid}| = n$
- **Worst case:** $|S_{mid}| = 1$



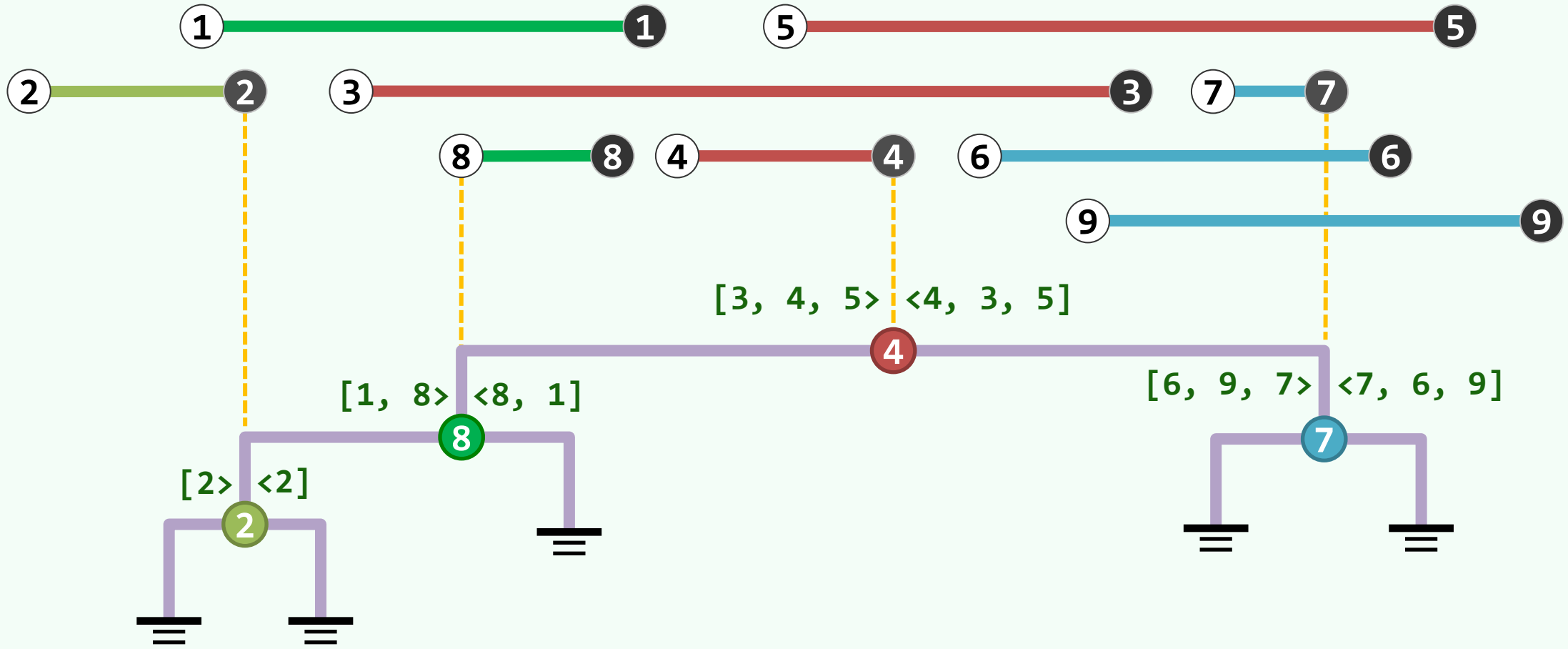
Associative Lists

❖ $L_{left/right}$ = all intervals of S_{mid} sorted by the LEFT/RIGHT endpoints



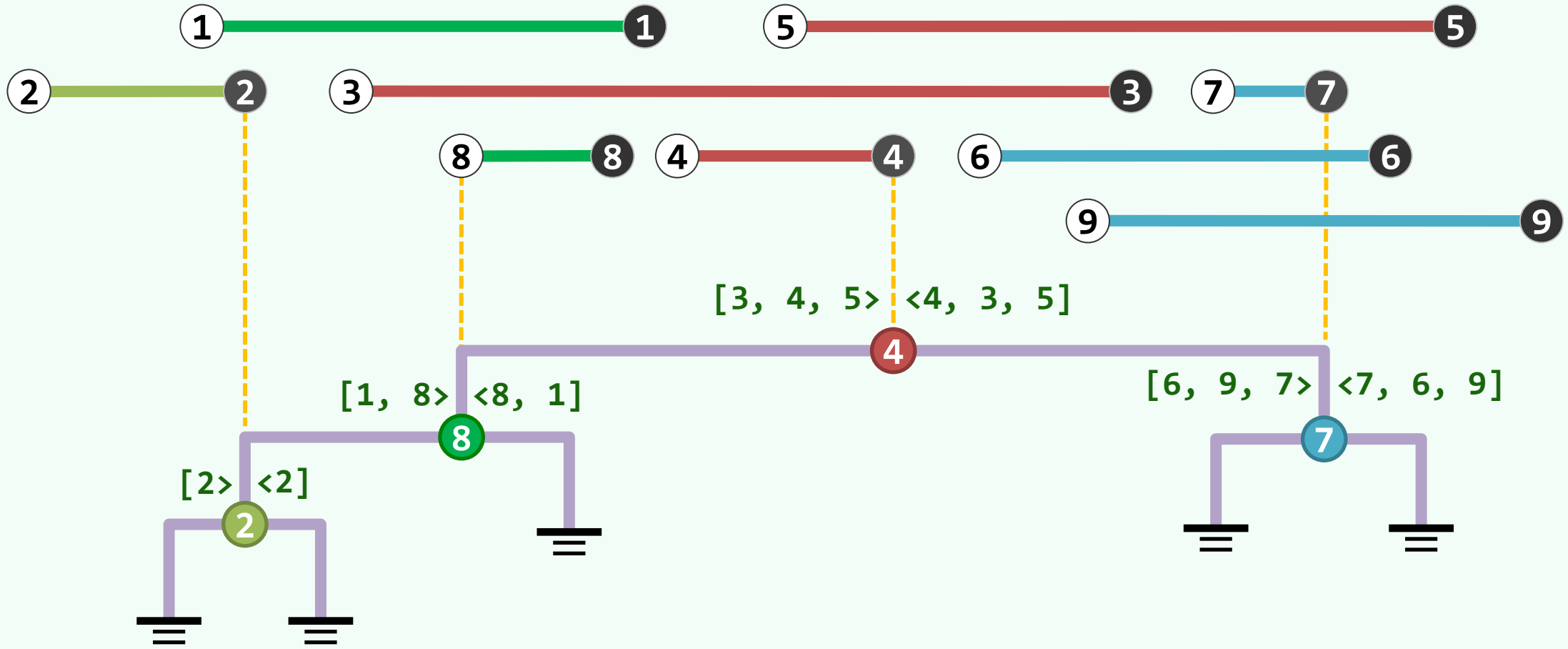
$\mathcal{O}(n)$ Size

❖ Each segment appears twice



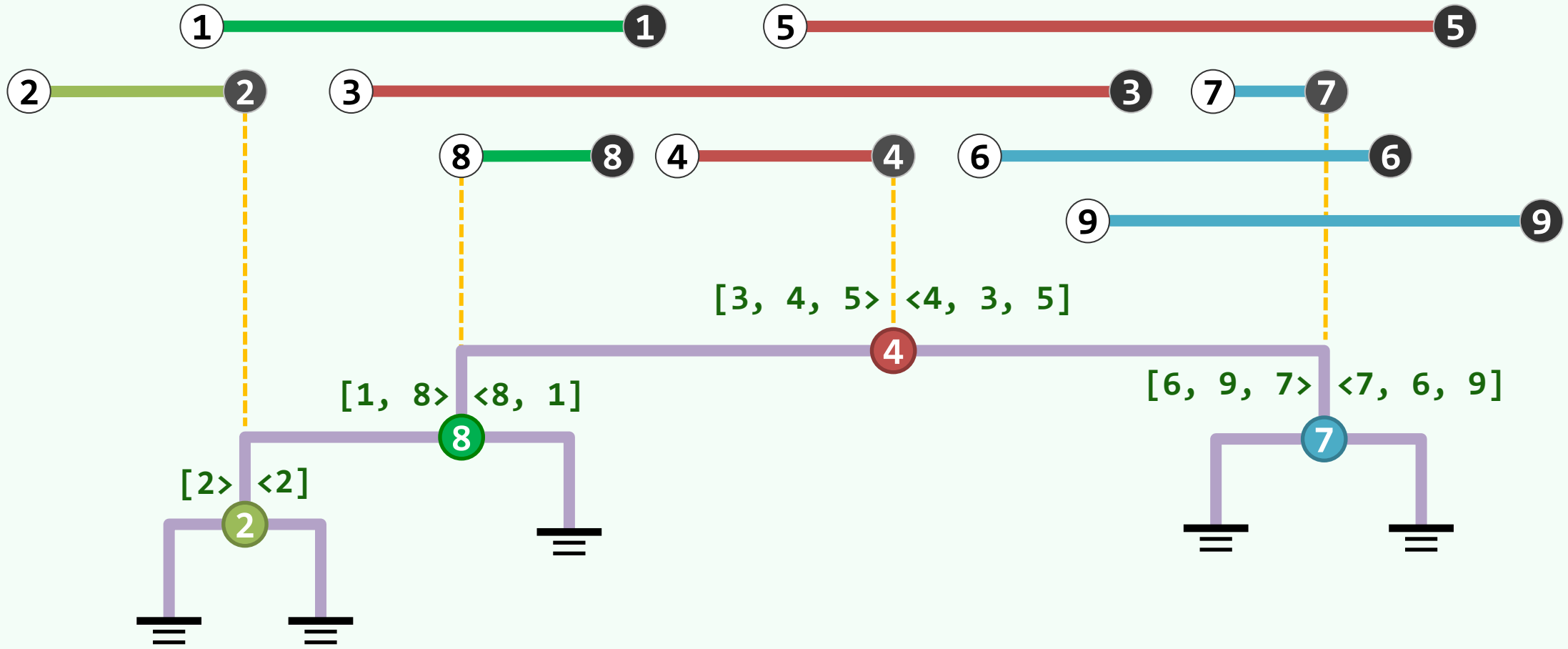
$\mathcal{O}(\log n)$ Depth

❖ Partitionings are done evenly



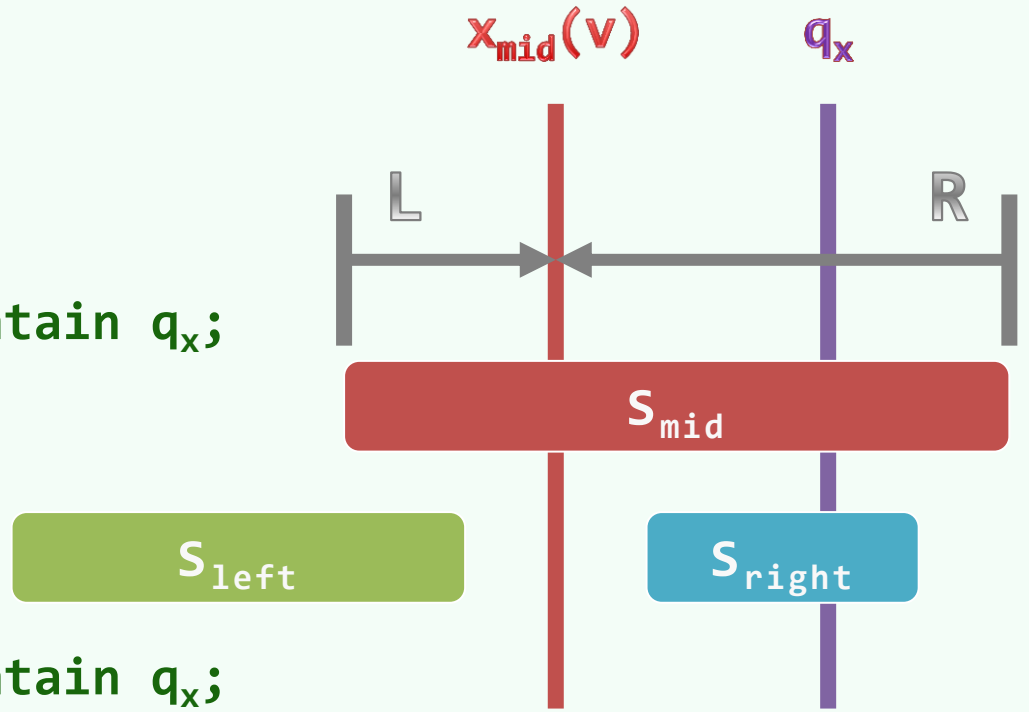
O(nlogn) Construction Time

❖ Hint: avoid repeatedly sorting



queryIntervalTree(v, q_x)

```
if ( ! v ) return;  //base
if ( q_x < x_mid(v) ) {
   report all segments of S_mid(v) that contain q_x;
   queryIntervalTree( lc(v), q_x );
} else if ( x_mid(v) < q_x ) {
   report all segments of S_mid(v) that contain q_x;
   queryIntervalTree( rc(v), q_x );
} else //with a probability ≈ 0
   report all segments of S_mid( v ); //both rc(v) & lc(v) can be ignored
```



O(r + logn) Query Time

❖ Each query visits O(logn) nodes //LINEAR recursion