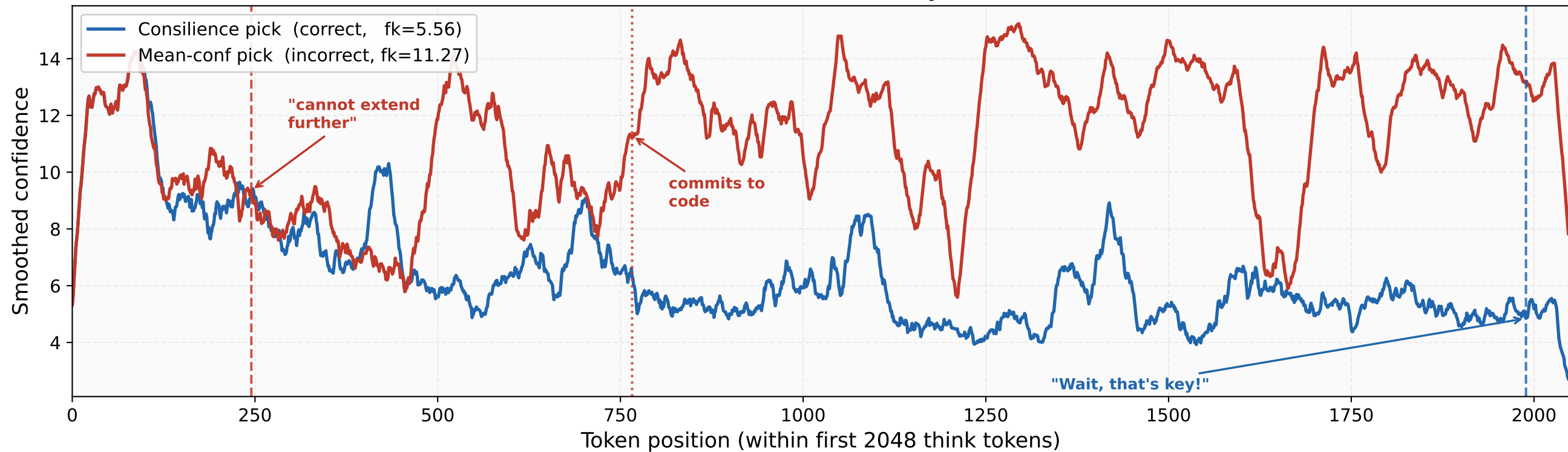


Problem 3 — LCB / Qwen3-80B (pass@1 = 0.03)



① 'stop on duplicate' framing — token ~245

the current path 4. When visiting a node: - If its value is already in the path, we cannot extend further (since values must be unique) - Otherwise, we can extend the path 5. Track the maximum path length found and the minimum number of nodes that achieve this maximum 6. For each valid path, calculate its total length (sum of edge weights)

Algorithm Details

1. Create adjacency list: for

② commits to implementation — token ~766 (fk=11.27)

Implementation

```
```python class Solution:    def longestSpecialPath(self, edges: List[List[int]],
nums: List[int]) -> List[int]: from collections import defaultdict
n = len(nums) # Build adjacency list graph
```

#### Consilience pick — reframing at token ~1989 (fk=5.56)

reverse from root to current node, we maintain a set of values in the current path. When we reach a node  $u$  with value  $\text{nums}[u]$ , if this value already exists in the current path, then we cannot have any path from any ancestor before the previous occurrence of this value to  $u$  (because the path would include two copies of  $\text{nums}[u]$ ). But we can have paths from ancestors after the previous occurrence of  $\text{nums}[u]$  to  $u$ .

Wait, that's key!

Let me think: if we have a path from root to  $u$ :  $[v_0, v_1, \dots, v_k=u]$  If  $\text{nums}[u]$  appears at position  $i$  and also at position  $j$  ( $j < k$ ), then any path that includes both  $v_i$  and  $v_k$  would have duplicate  $\text{nums}[u]$ . So the only valid paths ending at  $u$  are those that start from an ancestor after the last occurrence of  $\text{nums}[u]$  in the path from root to  $u$ .

Actually, no: if we