Minimum Size Subarray Sum

For this problem we need the shortest sub-array whose sum ≥ target.

|Approach 1: Sliding Window (O(n)):|

Key Idea:
- Use two pointers (start, end) to maintain a sliding window.
- Expand end to increase sum.
- Shrink start when sum ≥ target to minimize length.

|Algorithm:|

```
1. Initialize: · sum = 0
               · minLen = INT_MAX
               · start = 0
2. Loop end from 0 to n-1: · Add nums[end] to sum
                           · While sum >= target:
                                 · Update minLen = min(minLen, end-
                                   start+1).
                                 · Substract nums[start] and increment
                                   start.
3. Return minLen if found, else 0.
```

|Complexity:|

- Time: O(n) (each element visited at most twice)
- Space: O(1)

|Follow-up: O(n log n) Solution:|

Idea: Use prefix sums + binary search:
- Build prefix sum array.
- For each i, binary search for the smallest j such that prefix[j] - prefix[i] >= target.