Minimum Window Substring

For this problem we must find the smallest substring in s that contains all characters of t (including duplicates).

Approach: Sliding window + Hash Map; we use the two-pointer sliding window technique.

Steps:

1. Count frequency of characters in t using a hash map need.
2. Expand the window by moving right pointer:
   - Decrease need[s[right]] count.
   - If character count requirement met, increment formed.
3. When formed == required (all characters covered):
   - Try to shrink window by moving left pointer while still valid.
   - Update minimum length window whenever smaller valid window found.
4. Continue until right reaches end of s.

Complexity:

- Time: O(m+n) - each character processed at most twice (left/right pointers)
- Space: O(k) - where k is number of unique characters in t.