Longest Substring Without Repeating Characters

Solving the classic Longest Substring Without Repeating Characters problem - best tackled using the sliding window technique.

Strategy: Sliding Window with Hash Map:

1. Use a hash map to store the last seen index of each character.
2. Maintain a window [left, right] of unique characters.
3. For each character s[right]:
   - If it's already in the map and its last seen index ≥ left, move left to map[c]+1 to avoid duplicates.
   - Update the character's last seen index.
   - Update the max length: maxLen = max(maxLen, right - left + 1)

Time Complexity:
- O(n) - each character is visited at most twice (once by right, once by left)
- O(1) space if fixed alphabet (ASCII), else O(n)

Example Walkthrough:

For "pwwkew":
- p → window: p → length: 1
- w → pw → length 2
- another w → move left to skip first w → window: w continue
- k → wk → length 2
- e → wke → length 3 ✓