Optimized Architecture Plan

Divide and Conquer:

* Split the character set into segments.
* Use threads to handle each segment concurrently.

Incremental Hash Generation:

* Generate and test hashes in smaller increments.
* Use Breadth-First Search (BFS) to generate and check combinations level by level, ensuring lower memory usage.

Efficient Data Structures:

* Use a ConcurrentQueue to manage string combinations.
* Use a ConcurrentDictionary to track visited combinations.