Depth First Search

Project 8: Boggle Solver

(to understand recursion, first you need to understand recursion)
Seems like we’re missing a base case here ... and the progress case doesn’t seem to be making the problem smaller.
Let’s look at a simpler example ...
We can explore all 8 directions ...
The A tile is "in use." Don't use it again in the same word.
Column 4 is off the board. Don't go there.
Depth-first search logic

Given a position and a prefix ...
  If the position is off the board, do nothing
  If the position is already in use, do nothing
  New prefix = prefix + tile
  If it’s a complete word, note it
  If it’s a valid prefix
    Mark current tile as “in use”
    Recursively search in all 8 directions
    Unmark current tile before returning
/**
 * Continue search from board[row, col].
 * ... (fill in some documentation)
 */

private void search(int row, int col, String prefix) {

    ...

    String candidate = prefix + board.tileAt(row, col);
    Dict.Result match = dict.lookup(candidate);
    if (match == Dict.Result.WORD) {
        solutions.add(candidate);
    }

    ...

}
Summary

Depth-first search to find candidate words
Look each word up in dictionary
Base cases: Off board, already using that tile
Progress: Keep looking in the rest of the board, in any direction