**General idea introduction,**

Use backtracking algorithms:

Let the function dfs(i,j,k) indicate whether the word word[k...] can be searched from the (i,j) position of the grid. The word [k...] represents the suffix substring of the string word starting from the kth character.Return true if it can be found, false if not. The function dfs(i,j,k) is executed as follows:

• Boundary condition, if board [i] [j] ≠ word [k], the current character does not match, return false directly.

• If you have reached the end of the string and the corresponding character still matches, return true.

• Otherwise, traverse all adjacent locations of the current location. If you start from an adjacent location, you can search for the substring word[k+1...] , return true, otherwise return false.

In this way, we call dfs(i,j,0) at every position (i,j) to check: as long as one place returns True, it means that the corresponding word can be found in the grid, otherwise it means that it cannot be found.

To prevent repeated traversal of the same locations, an additional large visited array, such as board, is maintained to identify whether each location has been visited. Each time you traverse an adjacent location, you need to skip the location that has already been visited.