WordSearch.java

```
1     package com.example.trie;
2
3     import java.util.ArrayList;
4     import java.util.HashSet;
5     import java.util.List;
6     import java.util.Set;
7
8     class WordSearch {
9         Set<String> result = null;
10        char[][] board = null;
11        Trie1 trie = null;
12
13        public List<String> findWords(char[][] board, String[] words) {
14            this.board = board;
15            result = new HashSet<>();
16            // Build Trie1 on words
17            trie = new Trie1();
18            for (String word : words)
19 1              trie.add(word);
20            //////////
21
22            // Start recursion calls on each character in the board which contains entry in
23            // the root of the trie.
24 2          for (int i = 0; i < this.board.length; i++) {
25 2              for (int j = 0; j < this.board[i].length; j++) {
26 2                  if (trie.root.array[this.board[i][j] - 'a'] != null) {
27 2                      findWords(i, j, trie.root.array[this.board[i][j] - 'a'], new HashSet<String>());
28                    }
29                }
30            }
31 1          return new ArrayList<>(result);
32        }
33
34        void findWords(int i, int j, TrieNode curr, Set<String> visited) {
35 1          if (curr.value != null) {
36                result.add(curr.value);
37            }
38            visited.add(i + "#" + j);
39            TrieNode temp = null;
40            /*
41             * Conditional statements prunes invalid branches.
42             */
43 7          if (i > 0 && (temp = curr.array[board[i - 1][j] - 'a']) != null && !visited.contains((i - 1) + "#"
44 2              findWords(i - 1, j, temp, visited);
45            }
46
47 7          if (j > 0 && (temp = curr.array[board[i][j - 1] - 'a']) != null && !visited.contains(i + "#" + (j
48 2              findWords(i, j - 1, temp, visited);
49            }
50
51 7          if (i < board.length - 1 && (temp = curr.array[board[i + 1][j] - 'a']) != null
52 1                  && !visited.contains("" + (i + 1) + "#" + j)) {
53 2              findWords(i + 1, j, temp, visited);
54            }
55
56 7          if (j < board[i].length - 1 && (temp = curr.array[board[i][j + 1] - 'a']) != null
57 1                  && !visited.contains(i + "#" + (j + 1))) {
58 2              findWords(i, j + 1, temp, visited);
59            }
60            visited.remove(i + "#" + j);
61
62        }
63    }
64
65    class TrieNode {
66        TrieNode[] array;
67        String value;
68
69        TrieNode() {
70            array = new TrieNode[26];
71        }
72    }
73
74    class Trie1 {
```

```
75        TrieNode root;
76
77        Trie1() {
78            root = new TrieNode();
79        }
80
81        void add(String word) {
82            TrieNode temp = root;
83 2          for (int i = 0; i < word.length(); i++) {
84                char ch = word.charAt(i);
85 2              if (temp.array[ch - 'a'] == null) {
86 1                  temp.array[ch - 'a'] = new TrieNode();
87 1                  temp = temp.array[ch - 'a'];
88                } else {
89 1                  temp = temp.array[ch - 'a'];
90                }
91            }
92            temp.value = word;
93        }
94    }
```

Mutations

```
19  1. removed call to com/example/trie/Trie1::add → KILLED
    1. changed conditional boundary → KILLED
24  2. negated conditional → KILLED
    1. negated conditional → KILLED
25  2. changed conditional boundary → KILLED
    1. Replaced integer subtraction with addition → KILLED
26  2. negated conditional → KILLED
    1. Replaced integer subtraction with addition → KILLED
27  2. removed call to com/example/trie/WordSearch::findWords → KILLED
31  1. replaced return value with Collections.emptyList for com/example/trie/WordSearch::findWords → KILLED
35  1. negated conditional → KILLED
    1. negated conditional → KILLED
    2. Replaced integer subtraction with addition → SURVIVED
    3. negated conditional → NO_COVERAGE
43  4. negated conditional → KILLED
    5. Replaced integer subtraction with addition → NO_COVERAGE
    6. Replaced integer subtraction with addition → KILLED
    7. changed conditional boundary → KILLED
    1. removed call to com/example/trie/WordSearch::findWords → NO_COVERAGE
44  2. Replaced integer subtraction with addition → NO_COVERAGE
    1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer subtraction with addition → KILLED
    3. changed conditional boundary → KILLED
47  4. negated conditional → KILLED
    5. negated conditional → KILLED
    6. negated conditional → KILLED
    7. Replaced integer subtraction with addition → KILLED
    1. removed call to com/example/trie/WordSearch::findWords → KILLED
48  2. Replaced integer subtraction with addition → KILLED
    1. negated conditional → KILLED
    2. Replaced integer subtraction with addition → SURVIVED
    3. Replaced integer addition with subtraction → KILLED
51  4. negated conditional → KILLED
    5. Replaced integer addition with subtraction → KILLED
    6. Replaced integer subtraction with addition → KILLED
    7. changed conditional boundary → SURVIVED
52  1. negated conditional → KILLED
    1. removed call to com/example/trie/WordSearch::findWords → KILLED
53  2. Replaced integer addition with subtraction → KILLED
    1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer addition with subtraction → SURVIVED
    3. Replaced integer addition with subtraction → KILLED
56  4. negated conditional → KILLED
    5. negated conditional → KILLED
    6. Replaced integer subtraction with addition → KILLED
    7. changed conditional boundary → KILLED
57  1. negated conditional → KILLED
    1. Replaced integer addition with subtraction → KILLED
58  2. removed call to com/example/trie/WordSearch::findWords → KILLED
    1. changed conditional boundary → KILLED
83  2. negated conditional → KILLED
    1. Replaced integer subtraction with addition → KILLED
85  2. negated conditional → KILLED
86  1. Replaced integer subtraction with addition → KILLED
87  1. Replaced integer subtraction with addition → KILLED
89  1. Replaced integer subtraction with addition → NO_COVERAGE
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS

- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- com.example.trie.WordSearchTest.main(com.example.trie.WordSearchTest) (6 ms)

Report generated by PIT 1.15.0