419. Battleships in a Board

QuestionEditorial Solution
My Submissions

- Total Accepted: **5134**
- Total Submissions: **8560**
- Difficulty: **Medium**
- Contributors: **ben65**

Given an 2D board, count how many different battleships are in it. The battleships are represented with 'X's, empty slots are represented with '.'s. You may assume the following rules:

- You receive a valid board, made of only battleships or empty slots.
- Battleships can only be placed horizontally or vertically. In other words, they can only be made of the shape 1xN (1 row, N columns) orNx1 (N rows, 1 column), where N can be of any size.
- At least one horizontal or vertical cell separates between two battleships - there are no adjacent battleships.

Example:

```
X..X
...X
...X
```

In the above board there are 2 battleships.

Invalid Example:

```
...X
XXXX
...X
```

This is an invalid board that you will not receive - as battleships will always have a cell separating between them.

```
import java.util.Arrays;

public class BattleShip {


private static void dfs(char[][] board,int i,int j,boolean[][]
visited)
{
    int n = board.length;
    int m = board[0].length;
    visited[i][j]=true;
    //left
```

```
    if (i >= 0 && i < n && j - 1 >= 0 && j - 1 < m && !visited[i][j
- 1] && board[i][j - 1] == 'X')
    {
       dfs(board, i, j - 1,visited);
    }
    //right
    if (i >= 0 && i < n && j + 1 >= 0 && j + 1 < m && !visited[i][j
+ 1] && board[i][j + 1] == 'X')
    {
       dfs(board, i, j + 1,visited);
    }
    //top
    if (i - 1 >= 0 && i - 1 < n && j >= 0 && j < m && !visited[i -
1][j] && board[i - 1][j] == 'X')
    {
       dfs(board, i - 1, j,visited);
    }
    //bottom
    if (i + 1 >= 0 && i + 1 < n && j >= 0 && j < m && !visited[i +
1][j] && board[i + 1][j] == 'X')
    {
       dfs(board, i + 1, j,visited);
    }
}

public static int countBattleships(char[][] board) {
    int m = board.length;
    int n = board[0].length;
    boolean[][] visited = new boolean[m][n];
    for(int i=0;i<m;i++)
    Arrays.fill(visited[i], false);
    //System.out.printf("%b ", visited[2][2]);
    int num=0;
    for(int i=0;i<m;i++)
     for(int j=0;j<n;j++)
     {
       if(!visited[i][j] && board[i][j]=='X')
       {
          dfs(board, i, j, visited);
          num++;
       }
     }

     return num;
}


public static void main(String[] args) {
```

```java
// TODO Auto-generated method stub
/*
  X..X
  ...X
  ...X
 */
    char[][] board = {{'X','.','.','X'},
                      {'.','.','.','X'},
                      {'.','.','.','X'}};
    int ans = countBattleships(board);
    System.out.println(ans);
  }

}
```

Problems @ Javadoc Declaration Search

```
<terminated> BattleShip [Java Application] C:\Program
2
```