

Assignment no. 9

Title: Backtracking

You are given a 2D grid of letters (input 1) and a list of words. Find all words in a given list of words (input 2) that can be formed by following a path of adjacent letters (horizontally or vertically) in the grid.

Here's an example to illustrate the algorithm:

```
(input 1) Grid: [ ['A', 'B', 'C', 'E'],  
                  ['S', 'F', 'C', 'S'],  
                  ['A', 'D', 'E', 'E'] ]
```

```
(input 2) List of words: ["ABCCED", "SEE", "BFD"]
```

```
(output) Words found: ["ABCCED", "SEE"]
```

The algorithm starts at the first cell and tries to match it with the first letter of any word in the list. It finds a match with "ABCCED" and starts a recursive function with the current cell, the word, and an empty visited set. The function recursively explores neighboring cells until it matches the entire word. The algorithm then continues to the next cell and repeats the process until all cells have been explored. Any words that were found during the traversal are added to the list of valid words.

Solution: This is a classic problem known as the "Word Search Problem". A recursive backtracking algorithm that can solve this problem is as follows:

1. For each cell in the grid:

If the letter in the cell matches the first letter of any word in the list:

Start a recursive backtracking function with the current cell, the word, and a visited set.

2. In the recursive function:

1. If the word is empty, return True.

2. If the current cell is out of bounds or has already been visited, return False.

3. If the letter in the current cell does not match the next letter in the word, return False.

4. Mark the current cell as visited.

5. Recursively call the function with the neighboring cells and the remaining word.

6. If any of the recursive calls return True, return True.

7. Mark the current cell as unvisited and backtrack to the previous state.

3. Return a list of all words that were found.

The algorithm traverses the grid and tries to match the current cell with the first letter of any word in the list. If a match is found, it starts a recursive function to explore the neighboring cells and check if the remaining letters match any word in the list.