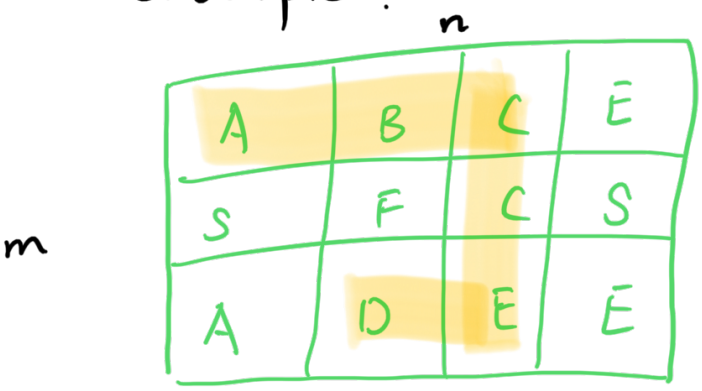
79. Word Search

- m x n grid

Example :



word: A B C C

Analyse :

based on above example :

start from A , possible directions :

A →

↓

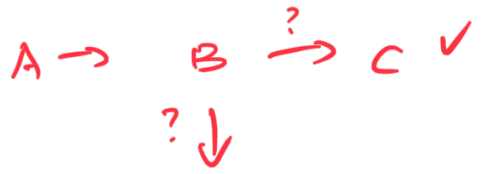
check whether there is B ,

if yes , then go that direction,

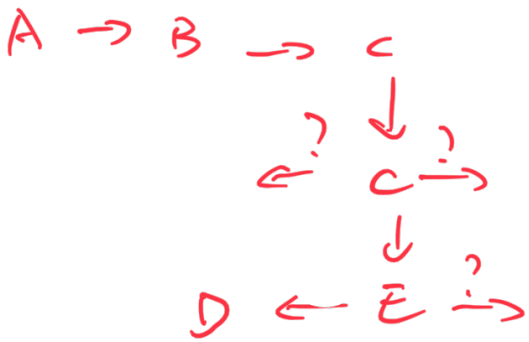
okay

A → B

possible directions for B:



Continue.



Finished.

As a result, we need to go through the 2D array to find the word [0]. then use DFS to go 4 directions to make the word

If Any direction of 4 directions does not work. We move on.

We need to mark the cell as

Visited when we decide to move from the cell to another, for avoiding visiting repeated place.

pseudo - code

```
def search:
        for i in m:
            for j in n:
                if (exist(...)):
                        return true

        return false.

def exist ( i, j, word, start, board )

    if start > len(word):
                return true

    if i<0 || i >= borad.len ||
       j<0 || j >= board[0].len:
                return false

    if board[i][j] == word[start]:
        start++ , c = board[i][j]
```

board [i] [j] = "#"

res = exist (i-1, j word, start, boar
|| exist (i+1, ----)
exist (i, j-1 ----)
exist (i, j+1 ----)

board [i] [j] = c

return res

return false.