DATE

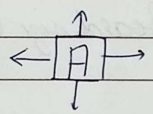
Backtracking III & Time Complexity Of Recursive Algorithm

Bad network will class

Q Word Search

A	B	C	E
S	F	C	S
A	D	E	E

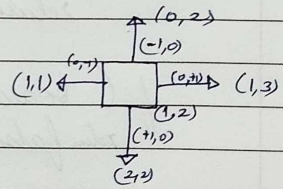
word = "ABCCED"



4 directions

→ Start from each cell

→ left $(x, y-1)$
right $(x, y+1)$
up $(x-1, y)$
down $(x+1, y)$



	0	1	2
0	B	A	B
1	L	A	B
2	V	R	O

word = "BABBAR"

(0,0) X RUD

(0,1) BXRUD

(0,2) XXXD

(1,2) LRUD

(2,2) BC → string match

boundary | visited | match

DATE

```
bool exist (vvchar &board, string word) {
    int m = board.size();
    int n = board[0].size();

    // call from each cell
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            if (solve (board, i, j, 0, word))
                return true;

    return false;
}

void solve (vvchar &board, int i, int j, int k, string& word){

    // if not safe
    if (i < 0 || i >= board.size())
        return false;

    if (j < 0 || j >= board[0].size())
        return false;

    if (board[i][j] == '—')
        return false;

    if (board[i][j] != word[k])
        return false;

    //BASE CASE
    if (k >= word.size() - 1) // last char reached & this
        return true;          // cond^n wasn't false
                              // so all good
```

classmate PAGE

DATE

```
char ch = board[i][j];

//mark visited ✓
board[i][j] = '-';

// traverse LRUD

bool left = solve(board, i, j-1, k+1, word);
— right = — (—, i, j+1, k+1, word);
— up = — (—, i-1, j, k+1, —)
— down = — (—, i+1, j, —, —)

board[i][j] = ch;

return left || right || up || down;
}
```

H/W Shorten call code using for loop.