



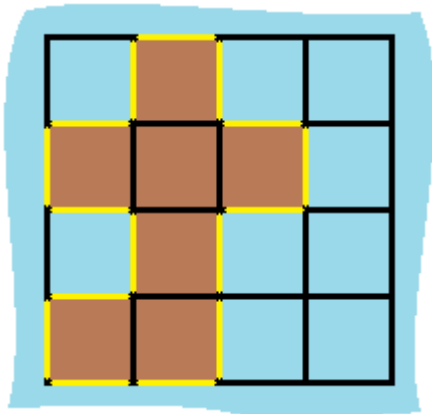
INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: MATRIZES COM E SEM RECURSÃO.

Prática 02

Parte 1: Matrizes sem Recursão

Problema 1: Resolva o problema [Crepúsculo em Portland \(Beecrowd 2168\)](#).

Problema 2: Resolva o problema [Island Perimeter \(LeetCode 463\)](#).



O problema consiste em, dada uma matriz representando um mapa com uma ilha, determinar o perímetro da ilha (destacadas na imagem). Há apenas uma ilha, as células que a compõe se conectam apenas verticalmente e horizontalmente, e não há lagos dentro da ilha.

Use o código abaixo para testar. Ao submeter a solução, copie e cole apenas a parte referente ao trecho destacado no código.

```
#include <iostream>
#include <vector>

using namespace std;

class Solution {
public:
    int islandPerimeter(vector<vector<int>>& grid) {
        ...
        return perimeter;
    }
};

int main() {
    vector<vector<int>> grid = {
        {0,1,0,0},
        {1,1,1,0},
        {0,1,0,0},
        {1,1,0,0}
    };

    cout << Solution().islandPerimeter(grid) << endl; // res = 16

    return 0;
}
```

Parte 2: Matrizes com Recursão

Problema 1: Estude o problema Chuva (OBI 2019), baixe o código ([github](#)) e estude-o. Faça ajustes que achar necessários para tornar o código mais fácil de entender. **Modifique** o código de forma que a chuva escorra apenas para um dos lados (esquerda ou direita), tanto na solução iterativa quanto na recursiva.

Problema 2: Resolva o problema [Contaminação \(Beecrowd 1583\)](#) adaptando o código e lógica usados no problema 1 (Chuva da OBI 2019). Adapte tanto a solução iterativa quanto a solução recursiva e teste ambas com as entradas de exemplo do Beecrowd.

Desafio (Opcional) 1: Resolva o problema [Fissura Perigosa \(OBI 2020 @ Neps.Academy\)](#).

Desafio (Opcional) 2: Resolva o problema [Ladrilhos \(Beecrowd 2246\)](#).