

Grid

Gustavo Silveira
Lucas Borges

O que é “Grid”?

- Pode ser entendido como uma classe de problemas que envolvem estruturas de dados de mais de uma dimensão, como por exemplo, matrizes.
- Temos como exemplo problemas mais simples como o Flood Fill, e também problemas mais complexos como String Matching 2D.

Flood Fill

- É geralmente encaixado em problemas de grafos, porém utiliza técnicas comuns em problemas que envolvem grid.
- Exemplo: Suponha que a matriz abaixo representa um parque, onde 'G' representa grama e 'A' representa água:

```
G G G G G A A A
G G G G A A A G
G G G G A A G G
G G G G G G G G
G G G A A A G G
G G G G G G G G
```

- Pergunta: Qual a maior área composta apenas de água sabendo que você só pode se mover nas direções Norte, Sul, Leste e Oeste?

Flood Fill

- Para resolvermos este problema é necessário percorrer a matriz, e para cada célula contabilizar o tamanho da área em que ela está contida.

```
int dr[] = {-1, 1, 0, 0};
int dc[] = {0, 0, -1, 1};
int vis[n][m];
int floodfill(int i, int j) {
    if (vis[i][j]) return 0;
    if (!val(i, j)) return 0;
    vis[i][j] = 1;
    int answer = 1;
    for (int i = 0; i < 4; ++i) {
        answer += floodfill(i + dr[i], j + dc[i]);
    }
    return answer;
}
```

```
bool val(int i, int j) {
    return (i >= 0 && i < n) &&
           (j >= 0 && j < m) &&
           grid[i][j] == 'A';
}
```

String matching 2D

- Outro problema que também é modelado como grid. Seu desafio é encontrar uma string (pattern) dentro de uma matriz de caracteres.
- Suponha a matriz abaixo:

abceefgh

bblamco

oprixasd

ureikjhpl

kadtrlmp

- Dado que somente é possível percorrer a matriz nas direções Norte, Sul, Leste, Oeste, Noroeste, Nordeste, Sudeste e Sudoeste, a palavra “abrir” existe no grid?

String matching 2D

- De forma semelhante ao problema apresentado anteriormente, podemos fazer uma recursão para percorrer a matriz seguindo os vetores de direção.

```
int dr[] = {-1, 1, 0, 0, -1, -1, 1, 1}; //agora temos 8 opções
int dc[] = {0, 0, -1, 1, 1, -1, 1, -1};
int vis[n][m]; char grid[n][m]; string pattern;
int find(int i, int j, int pos) {
    if (pos == (int)pattern.size()) return 1;
    if (!val(i, j)) return 0;
    if (vis[i][j] != -1) return vis[i][j];
    vis[i][j] = 0;
    if (grid[i][j] == pattern[pos]) {
        for (int i = 0 ; i < 8; ++i) {
            vis[i][j] = vis[i][j] || find(i + dr[i], j + dc[j], pos + 1);
            if (vis[i][j]) return 1;
        }
    }
    return 0; }
```

Obrigado