

ESCOLA DE PRIMAVERA DA MARATONA SBC DE PROGRAMAÇÃO



PROMOÇÃO:



APOIO:



Grupo de Computação Competitiva

BUSCA EM LARGURA E FLOOD FILL



Por: *Gabriela Emanuele de Araujo Amorim*

CONTEÚDOS

- 01 - Problema motivador (Coloração de Cenários de Jogos - 1907)
- 02 - Analisando o Problema
- 03 - Busca em Largura (BFS)
- 04 - Funcionamento do algoritmo
- 05 - Algoritmo
- 06 - Flood Fill
- 07 - Resolução do problema
- 08 - Problemas

01 - PROBLEMA MOTIVADOR

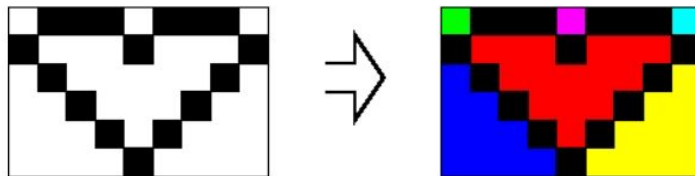
beecrowd | 1907

Coloração de Cenários de Jogos

Por Leandro Zatesko, UFFS 🇧🇷 Brazil

Timelimit: 1

O Prof. Fernando Bevilacqua está muito preocupado com os cenários do seu mais novo jogo. Os contornos dos cenários já foram desenhados por um artista, restando ao Prof. Fernando apenas colori-los. No momento, cada cenário é uma imagem em que cada *pixel* está preto ou branco. Assim, quando o Prof. Fernando, em seu programa de coloração de imagens, clica num *pixel* branco para ser colorido com uma cor α , toda a *região branca* em que está o *pixel* selecionado recebe a cor α . Dizemos que um *pixel* branco **A** está na mesma região branca que um *pixel* branco **B** se existe um caminho entre **A** e **B** que passa apenas por *pixels* brancos e que considera as adjacências apenas nos sentidos horizontal e vertical. Por exemplo, são necessários 6 cliques para colorir a figura da esquerda.



Fonte: <https://judge.beecrowd.com/pt/problems/view/1907>

01 - PROBLEMA MOTIVADOR

Entrada

A primeira linha da entrada consiste de dois inteiros positivos **N** e **M** ($N, M \leq 1.024$), os quais representam a *resolução* da imagem. Cada uma das **N** linhas seguintes contém **M** caracteres, os quais podem ser **.** (ponto) ou **o** (letra 'o' minúscula), representando respectivamente um *pixel* branco ou um *pixel* preto.

Saída

Imprima uma linha contendo um único inteiro que represente o número de cliques necessários para colorir toda a figura descrita na entrada.

Exemplos de Entrada	Exemplos de Saída
<pre>6 9 .ooo.ooo. o...o...o .o....o. ..o...o.. ...o.o... ...o....</pre>	6
<pre>1 8 .o.o.o.o</pre>	4
<pre>1 1 o</pre>	0

Fonte: <https://judge.beecrowd.com/pt/problems/view/1907>

02 - ANALISANDO O PROBLEMA



Resposta: 6

03 - BUSCA EM LARGURA (BFS)

A Busca em Largura (BFS) explora um grafo expandindo todos os vértices de uma mesma distância antes de passar para a próxima. Ela encontra o menor caminho (em número de arestas) de um vértice de origem para os outros vértices acessíveis.

- **Processo:** Cada vértice pode estar em três estados:
 - Não visitado;
 - Processo de exploração;
 - Totalmente processado.
- **Objetivo:** Descobrir todos os vértices acessíveis e calcular a distância mínima até eles.

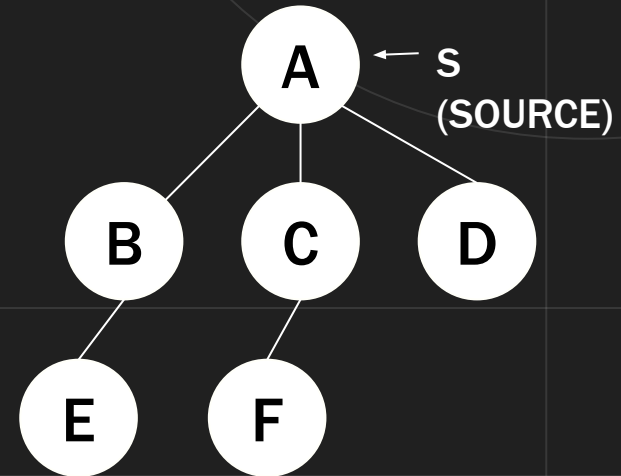
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.

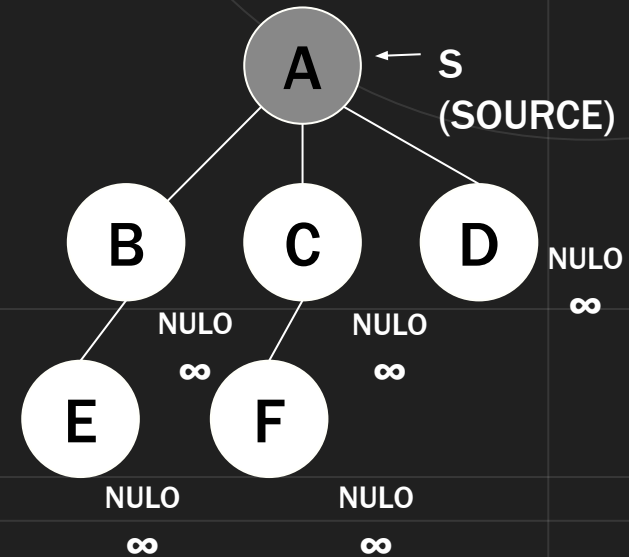


04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 
  
```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



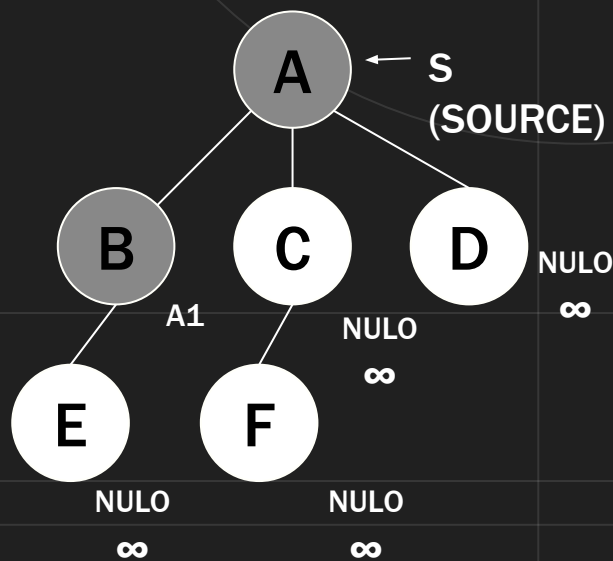
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



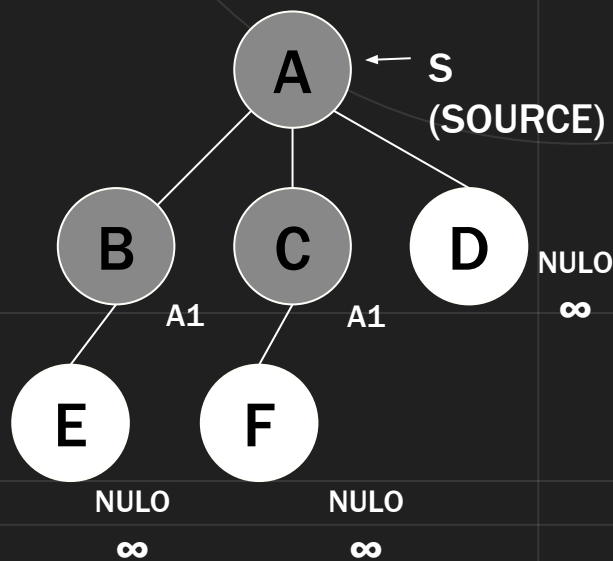
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



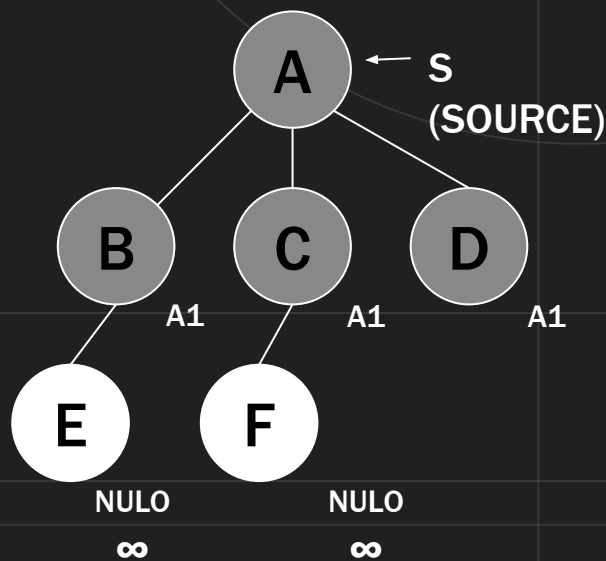
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



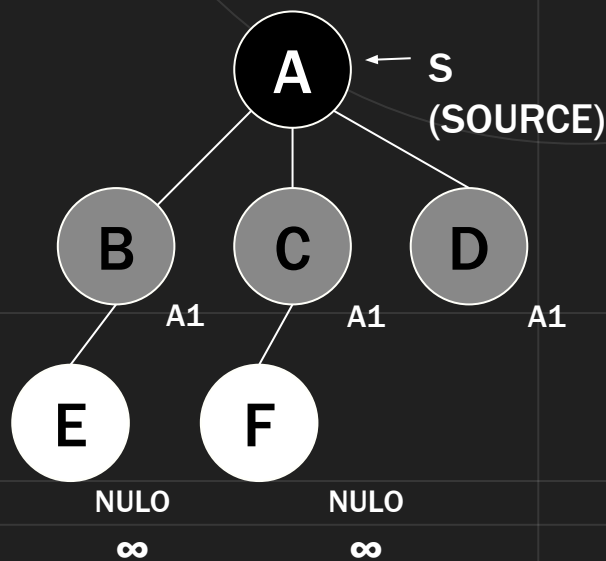
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



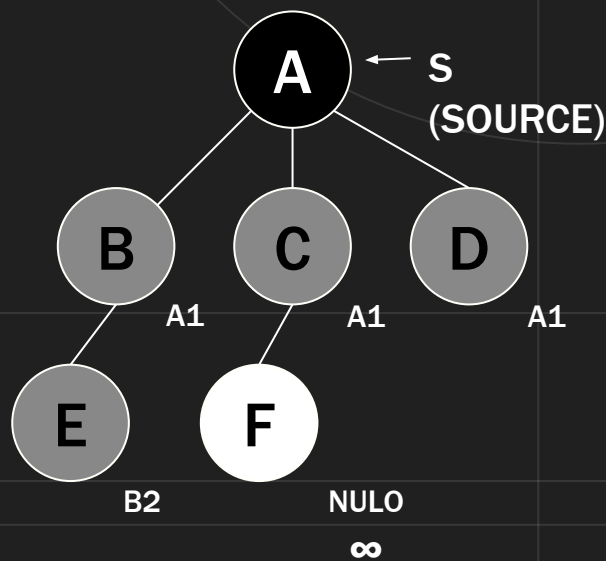
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



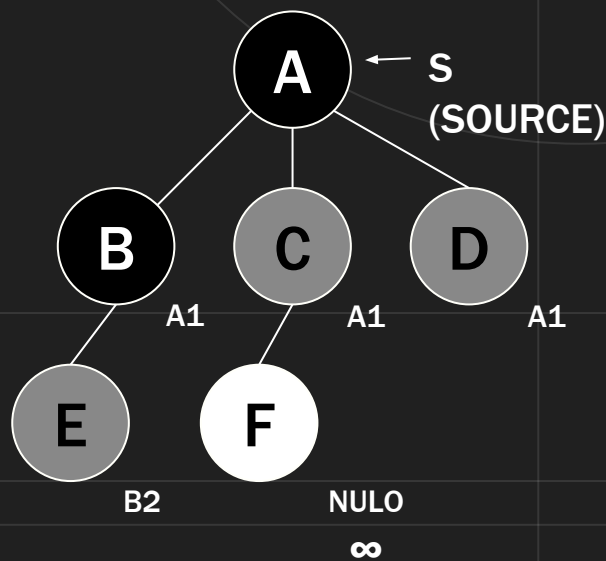
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



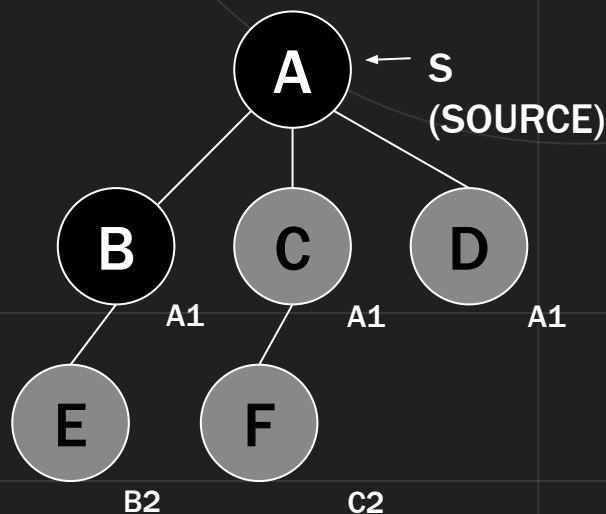
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



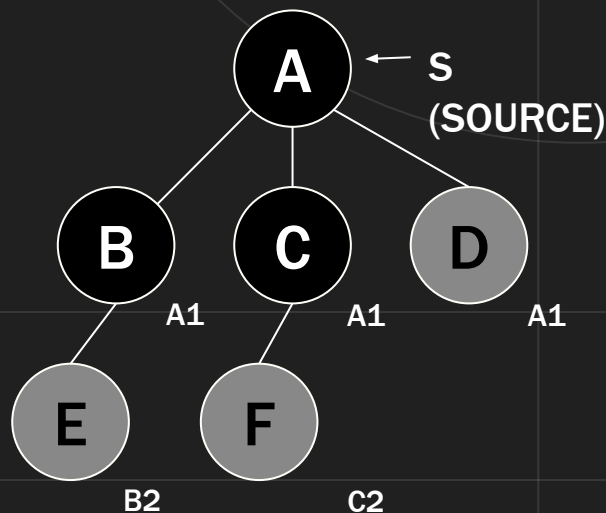
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



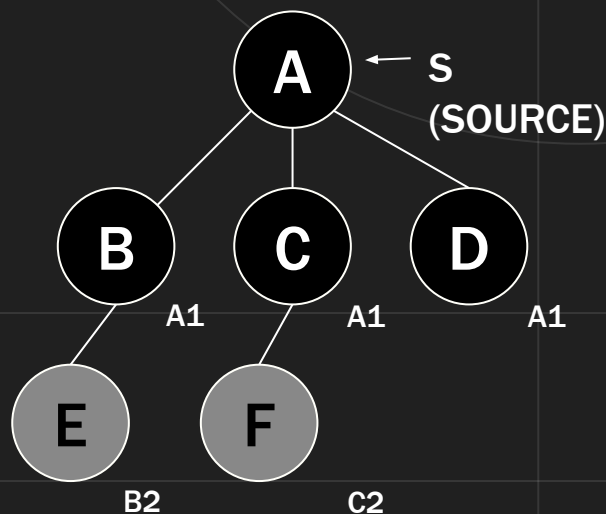
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



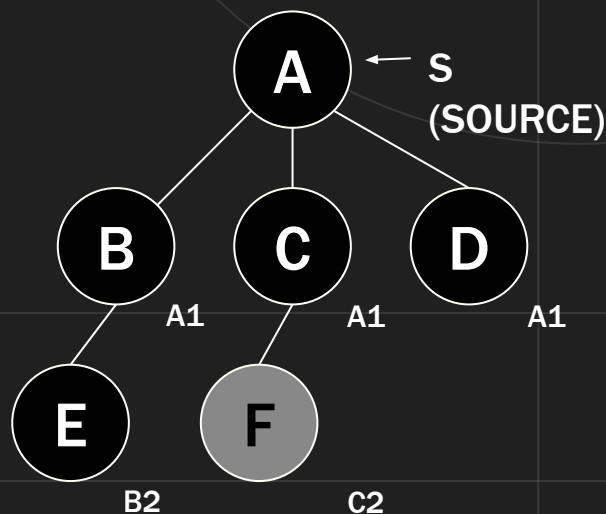
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



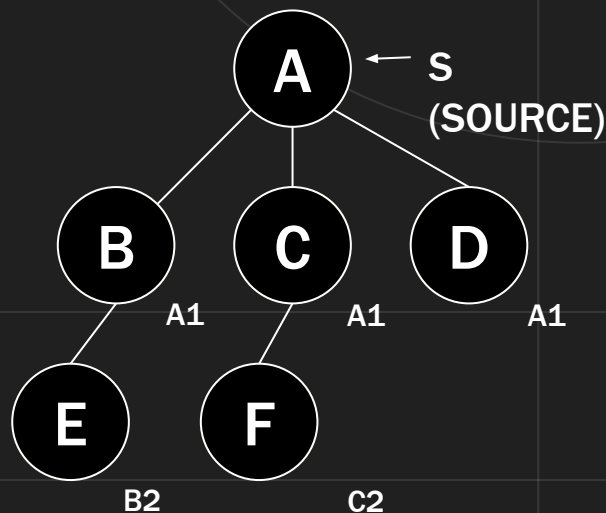
04 - FUNCIONAMENTO DO ALGORITMO

```

1  for cada vértice  $u \in V[G] - \{s\}$ 
2    do  $cor[u] \leftarrow \text{BRANCO}$ 
3     $d[u] \leftarrow \infty$ 
4     $\pi[u] \leftarrow \text{NULO}$ 
5   $cor[s] \leftarrow \text{CINZA}$ 
6   $d[s] \leftarrow 0$ 
7   $\pi[s] \leftarrow \text{NULO}$ 
8   $Q \leftarrow \emptyset$ 
9  ENQUEUE(Q, s)
10 while  $Q \neq \emptyset$ 
11   do  $u \leftarrow \text{DEQUEUE}(Q)$ 
12   for cada  $v \in \text{Adj}[u]$ 
13     do if  $cor[v] = \text{BRANCO}$ 
14       then  $cor[v] \leftarrow \text{CINZA}$ 
15          $d[v] \leftarrow d[u] + 1$ 
16          $\pi[v] \leftarrow u$ 
17         ENQUEUE(Q, v)
18    $cor[u] \leftarrow \text{PRETO}$ 

```

Fonte: CORME, Thomas. *Algoritmos Teoria e Prática*. [S. l.: s. n.], 2012.



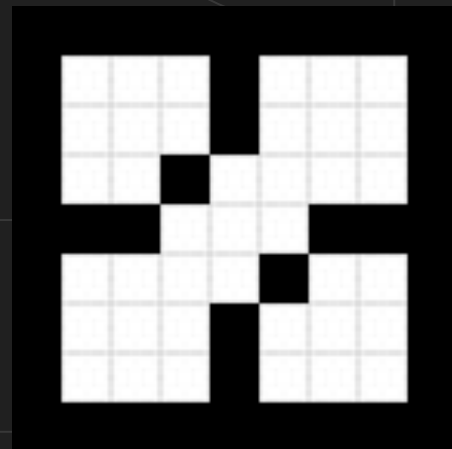
05 - ALGORITMO

06 - FLOOD FILL

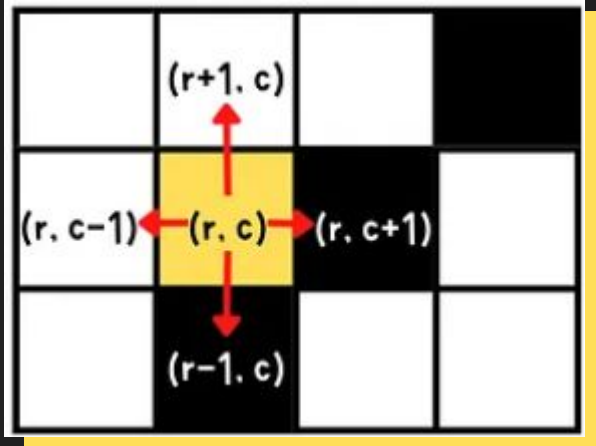
- Algoritmo usado para determinar uma área conectada em uma matriz bidimensional;
- Simula o comportamento da ferramenta "balde de tinta" em editores de imagem.

Funcionamento:

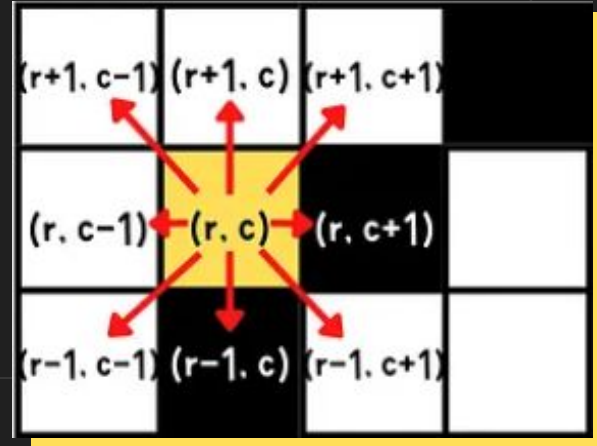
1. Ponto de Partida: Começa em uma posição inicial (x, y).
2. Direções: Decide se vai em 4 ou 8 direções.
3. Verificação:
 - a. Se a célula tem a cor de destino.
 - b. Substitui pela cor substituta.
4. Recursão/Iteração: Repete o processo até preencher a área conectada.



06 - FLOOD FILL



Four-way



Eight-way

06 - FLOOD FILL (ALGORITMO)

```
1 flood_fill(x, y, check_validity)
2   Queue q
3   q.push((x, y))
4   while (q is not empty)
5       (x1, y1) = q.pop()
6       color(x1, y1)
7       if (check_validity(x1 + 1, y1))
8           q.push(x1 + 1, y1)
9       if (check_validity(x1 - 1, y1))
10          q.push(x1 - 1, y1)
11      if (check_validity(x1, y1 + 1))
12          q.push(x1, y1 + 1)
13      if (check_validity(x1, y1 - 1))
14          q.push(x1, y1 - 1)
```




07 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

Estratégia:

- Utilizar o algoritmo Flood Fill para contar regiões conectadas de pixels brancos;
- Pixels brancos conectados ortogonalmente (esquerda, direita, cima, baixo) formam uma única região.

Solução:

- Percorrer a matriz;
- Ao encontrar um pixel branco (.), iniciar o Flood Fill (BFS) para marcar toda a região conectada;
- Cada novo Flood Fill indica um novo clique;
- Contar o número de cliques realizados.

07 - RESOLUÇÃO DO PROBLEMA MOTIVADOR

08 - PROBLEMAS

Duende Perdido - <https://judge.beecrowd.com/pt/problems/view/2294>

Colorindo - <https://judge.beecrowd.com/pt/problems/view/2405>

Manchas de Pele - <https://judge.beecrowd.com/pt/problems/view/3061>

REFERÊNCIAS

[1] CORME, Thomas. **Algoritmos Teoria e Prática**. [S. l.: s. n.], 2012.

[2] FLOOD/SEED Fill Algorithm. [S. l.], 2022. Disponível em: <https://yuminlee2.medium.com/flood-seed-fill-algorithm-21fba08a46e>. Acesso em: 10 out. 2024.

OBRIGADO PELA ATENÇÃO

Grupo de Computação Competitiva

