

1 Infográfico

1.1 Coloração sequencial de grafos

A convenção do uso de cores em grafos se origina no seu uso para a coloração de países em mapas. Em representações matemáticas e computacionais, é comum usar os primeiros números inteiros positivos ou não negativos como as cores. Em geral pode-se usar qualquer conjunto finito de cores. A natureza do problema depende da quantidade das cores, e não da sua natureza.

2 Livro: Ordenação de elementos em grafos

Uma aresta direcionada de um grafo é um par ordenado de vértices (u, v) representando uma conexão assimétrica de u a v . As arestas normais podem ter seu comportamento repetido por meio de arestas bidirecionais, ou de duas arestas direcionais, cada uma delas ligando pro o outro elemento.

Um grafo direcionado e sem ciclos é chamado de **grafo acíclico direcionado**, e podem também ser chamados de DAGs. Eles possuem muitos recursos que os tornam mais simples que os grafos. Para começar, sempre são dotados de vértices conhecidos como fontes e sumidouros. Uma **fonte** é um vértice sem aresta de entrada; um **sumidouro** é um vértice sem saída.

2.1 Ordenação topológica

A ordenação topológica funciona como uma ordenação natural das DAGs. E ela dá suporte a um recurso visual de disposição de vértices e arestas do grafo. Para isso, é preciso listar todas as fontes da esquerda para a direita e representar as arestas direcionadas de G entre essas fontes. Na verdade, qualquer lista de vértices é uma ordenação topológica se todas as arestas estiverem direcionadas até o fim da lista.

2.1.1 Algoritmo de ordenação Topológica

Um dos algoritmos comumente empregados para a obtenção de uma ordem topológica de um grafo faz uso de um procedimento de varredura, que o algoritmo de **busca em profundidade**. Esse algoritmo possui uma estrutura de dados auxiliar para armazenar os vértices na ordem em que a visitação de suas vizinhanças é finalizada.

Algoritmo 1: Busca em profundidade (BP)

```
tempo  $\leftarrow$  0 pilhaVertices  $\leftarrow$  para cada  $v \in V(G)$  faça se  $v.status = \text{NOVO}$ 
entao visitaBP( $v$ , listaVertices) fim se,para retorna listaVertices para cada
 $v \in V(G)$  faça  $v.status \leftarrow \text{NOVO}$  fim para
tempo  $\leftarrow$  0 pilhaVertices  $\leftarrow$  para cada  $v \in V(G)$  faça se  $v.status$ 
= NOVO entao visitaBP( $v$ , listaVertices) fim se,para retorna listaVertices
```