

BUSCA EM PROFUNDIDADE

PROJETO E ANÁLISE DE ALGORITMOS, DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO – UFPI

MARCELO VICTOR SOUSA LIMA, SEBASTIÃO PEREIRA MENDES NETO.



INTRODUÇÃO – DEFINIÇÃO DE GRAFOS

- Um grafo $G = (V, A)$ é constituído de um conjunto de vértices e um conjunto de arestas que conectam pares de vértices.
- O grafo G pode ser direcionado, não direcionado, ponderado entre outras classificações.
- Um grafo pode ser representado por uma matriz de adjacências ou por uma lista de adjacências.
- Grafos são bastante úteis na computação, física, matemática e engenharia. Para problemas que vão desde a saída de um labirinto ao estudo de uma rede social.

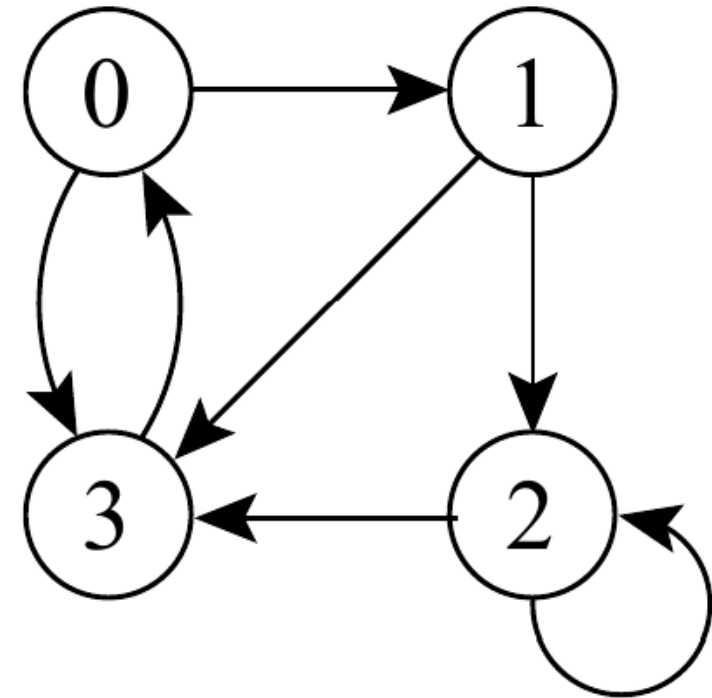


Figura 1: Exemplo de gráfico direcionado

BUSCA EM PROFUNDIDADE - DEPTH-FIRST SEARCH (DFS)

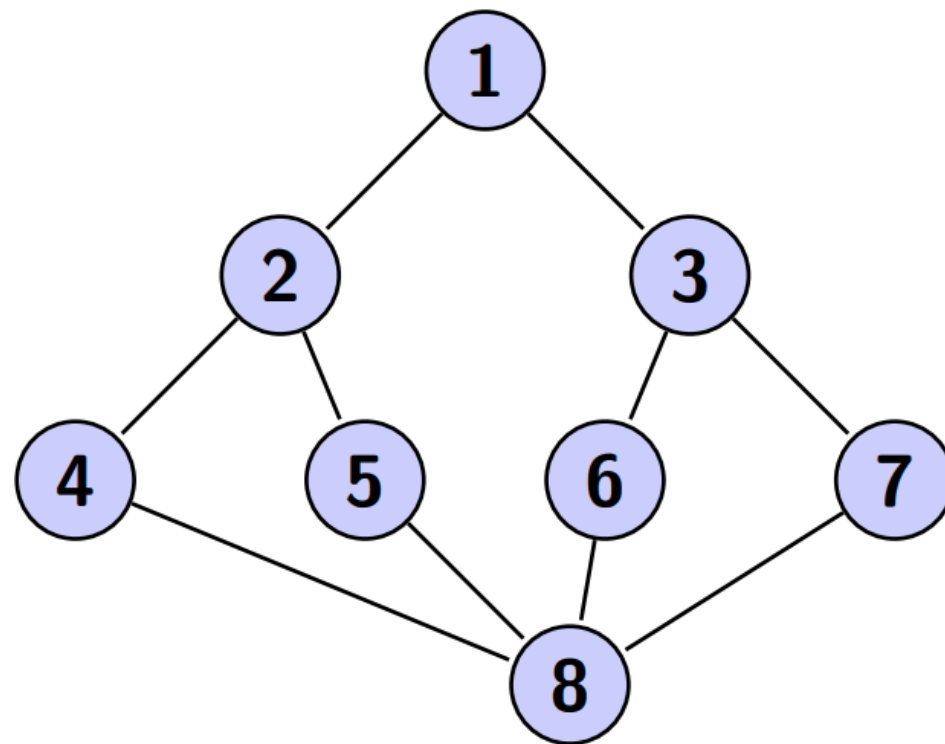
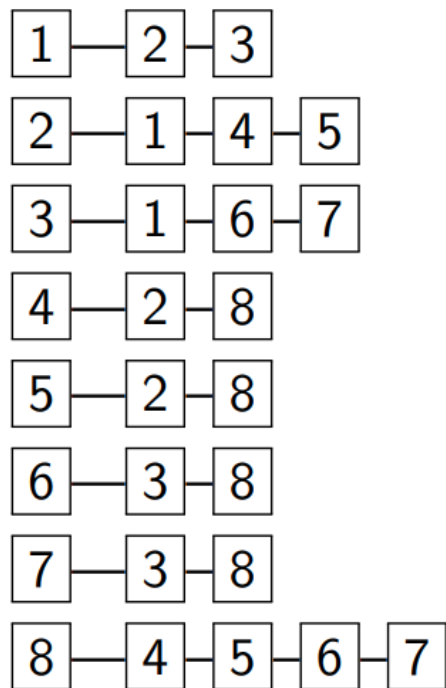
- A busca em profundidade (DFS) é um algoritmo para caminhar no grafo.
- Serve para encontrar a saída de um labirinto.
- A estratégia é buscar o mais profundo possível no grafo sempre que possível.
- As arestas são exploradas a partir do vértice u mais recentemente descoberto que ainda possui arestas não exploradas saindo dele.
- Quando todas as arestas adjacentes a u tiverem sido exploradas a busca anda para trás para explorar vértices que saem do vértice do qual u foi descoberto.

BUSCA EM PROFUNDIDADE - DEPTH-FIRST SEARCH (DFS)

- Para acompanhar o progresso do algoritmo cada vértice é colorido de branco, cinza ou preto.
- Todos os vértices são inicializados branco.
- Quando um vértice é descoberto pela primeira vez ele torna-se cinza, e é tornado preto quando sua lista de adjacentes tenha sido completamente examinada.
- Onde $u.d$ representa quando o vértice é descoberto pela primeira vez. Colorindo de cinza.
- Enquanto $u.f$ registra quando a busca termina de analisar a lista de adjacência. Colorindo de preto.

BUSCA EM PROFUNDIDADE - EXEMPLO

Listas de adjacências:



PSEUDOCÓDIGO DO DFS

DFS(G)

```
1  for each vertex  $u \in G.V$ 
2       $u.color = \text{WHITE}$ 
3       $u.\pi = \text{NIL}$ 
4   $time = 0$ 
5  for each vertex  $u \in G.V$ 
6      if  $u.color == \text{WHITE}$ 
7          DFS-VISIT( $G, u$ )
```

DFS-VISIT(G, u)

```
1   $time = time + 1$                                 // white vertex  $u$  has just been discovered
2   $u.d = time$ 
3   $u.color = \text{GRAY}$ 
4  for each vertex  $v$  in  $G.Adj[u]$  // explore each edge  $(u, v)$ 
5      if  $v.color == \text{WHITE}$ 
6           $v.\pi = u$ 
7          DFS-VISIT( $G, v$ )
8   $time = time + 1$ 
9   $u.f = time$ 
10  $u.color = \text{BLACK}$                                 // blacken  $u$ ; it is finished
```

ANÁLISE DE COMPLEXIDADE DO DFS

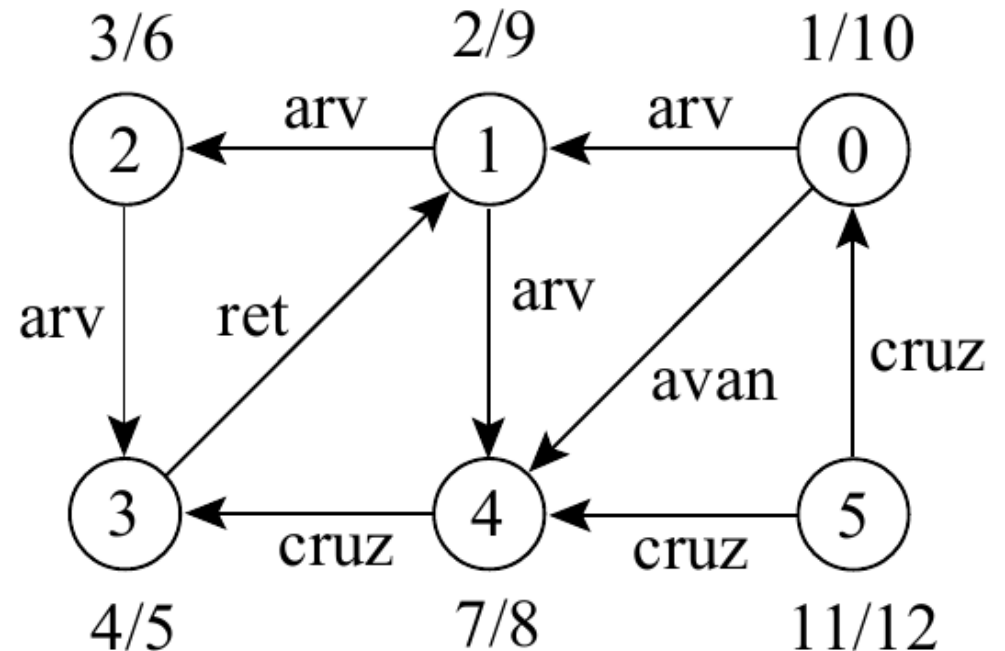
- Os loops das linhas 1-3 e 5-7 da DFS levam $\Theta(|V|)$.
- A função DFS-VISIT é chamada exatamente uma vez para cada vértice $v \in V$ considerando que o vértice u seja branco.
- Quando a DFS-VISIT é chamada, ela altera a cor do vértice u para cinza.
- O loop nas linhas 4-7 do DFS-VISIT executam $|Adj[u]|$ vezes.
 - Portanto: $\sum_{u \in V} |Adj[u]| = \Theta(|A|)$
- O custo total portanto é $\Theta(|V|) + \Theta(|A|) = \Theta(|V| + |A|)$.

CLASSIFICAÇÃO DE ARESTAS

- **Arestas de árvores:** são arestas de uma árvore de busca em profundidade. A aresta (u, v) é uma aresta de árvore se v foi descoberto pela primeira vez ao percorrer a aresta (u, v) .
- **Arestas de retorno:** conectam um vértice u com um antecessor v em uma árvore de busca em profundidade (inclui self-loops).
- **Arestas de avanço:** não pertencem à árvore de busca em profundidade mas conectam um vértice a um descendente que pertence à árvore de busca em profundidade.
- **Arestas de cruzamento:** podem conectar vértices na mesma árvore de busca em profundidade, ou em duas árvores diferentes.

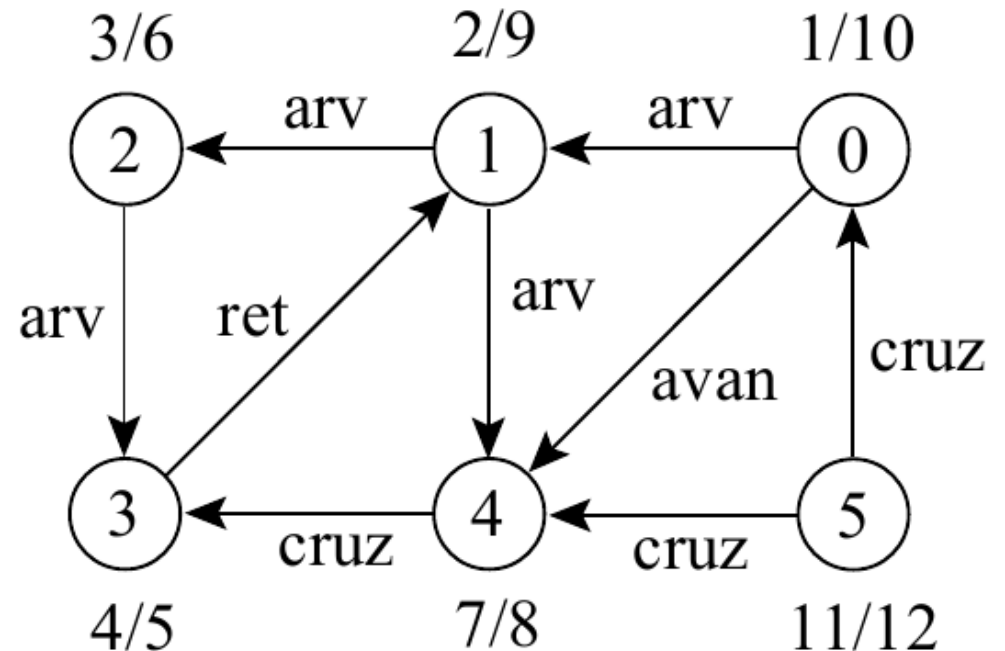
CLASSIFICAÇÃO DE ARESTAS - COLORAÇÃO

- Na busca em profundidade cada aresta pode ser classificada pela cor do vértice que é alcançado pela primeira vez
- Branco indica uma aresta da árvore;
- Cinza indica uma aresta de retorno;
- Preto indica uma aresta de avanço quando u é descoberto antes de v ou uma aresta de cruzamento caso contrário.



CLASSIFICAÇÃO DE ARESTAS - COLORAÇÃO

- Na busca em profundidade cada aresta pode ser classificada pela cor do vértice que é alcançado pela primeira vez
- Branco indica uma aresta da árvore;
- Cinza indica uma aresta de retorno;
- Preto indica uma aresta de avanço quando u é descoberto antes de v ou uma aresta de cruzamento caso contrário.



REFERÊNCIAS

- CORMEN, Thomas et al. Algoritmos -Teoria e Prática (3ª edição). **Editora Campus**, 2012.
- ZIVIANI, N. Projeto de Algoritmos: com implementações em Pascal e C, 2ª edição, **Cengage Learning**, 2009.
- <https://www.ic.unicamp.br/~rocha/teaching/2014s1/mc202/aulas/aula-grafos.pdf>
- [GGI026 - Grafos - Busca em largura e profundidade \(ufu.br\)](#)



OBRIGADO