

Teoria dos Grafos
Introdução
Busca em Grafos

Prof. Humberto Brandão
humberto@unifal-mg.edu.br

aula disponível no site:
<http://bcc.unifal-mg.edu.br/~humberto/>

Universidade Federal de Alfenas
Departamento de Ciências Exatas
versão da aula: 0.2

Busca em Grafos

- Os **objetivos** da busca são:
 - determinar quais **vértices** são **alcançáveis** através de um vértice inicial...
 - Se um determinado objeto está presente no grafo...
- Aplicações???

Busca em Grafos

- Aplicações:
 - Compiladores;
 - Resolução de problemas (xadrez, por exemplo);
 - Função “localizar arquivo” no sistema operacional;
 - Detecção de *deadlocks*.
 - Dentre centenas de outras...

Busca em Grafos

- Adaptações nos algoritmos de busca nos permitem construir algoritmos para os problemas de:
 - Árvore Geradora Mínima (AGM);
 - Caminho Mínimo;
 - Componentes Fortemente Conectados;
 - Ordenação Topológica.

Busca em Grafos

- Algoritmos clássicos de Busca:
 - Busca em Largura;
 - Busca em Profundidade;
 - *Backtracking*;
 - Adaptação da busca em profundidade;
 - *Backtracking* com *backjumping*;
 - Idem.

Busca em Profundidade

Busca em Profundidade

- A **busca em profundidade** (do inglês *depth-first search* - **DFS**) é um algoritmo para caminhar no grafo;
- Seu núcleo se **concentra em buscar**, sempre que possível, **o mais fundo no grafo**.
- As arestas são exploradas a partir do vértice v mais recentemente descoberto que ainda possui arestas não exploradas saindo dele.

Busca em Profundidade

- Quando todas arestas adjacentes a v tiverem sido exploradas, a busca "anda para trás" (do inglês *backtrack*) para explorar vértices do qual v foi descoberto;
- O processo continua até que sejam descobertos todos os vértices que são alcançáveis a partir do vértice original;
- Se todos os vértices já foram descobertos, então é o fim.
- Caso contrário o processo continua a partir de um novo vértice de origem ainda não descoberto (grafos desconexos).
 - Este é um ponto diferenciado da busca em árvore que vocês já conhecem;
 - Pois ao final de uma busca simples, pode haver vértices que não foram alcançados.

Busca em Profundidade

- Legenda para algoritmo:
 - Vértice Branco** – Ainda não visitado...
 - Vértice cinza** – Visitado, mas seus adjacentes ainda não foram todos visitados;
 - Vértice preto** – Visitado, e seus adjacentes já foram todos visitados.

Busca em Profundidade

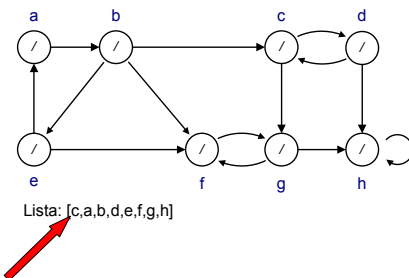
DFS(G) 1 para cada vértice $u \leftarrow V[G]$ 2 $cor[u] \leftarrow BRANCO$ 3 $tempo \leftarrow 0$ 4 para cada vértice $u \in V[G]$ 5 se $cor[u] = BRANCO$ 6 DFS-VISIT(u)	DFS-VISIT(u) 1 $cor[u] \leftarrow CINZA$ 2 $tempo \leftarrow tempo + 1$ 3 $d[u] \leftarrow tempo$ 4 para cada vértice $v \in Adj(u)$ 5 se $cor[v] = BRANCO$ 6 DFS-VISIT(v) 7 $cor[u] \leftarrow PRETO$ 8 $f[u] \leftarrow tempo \leftarrow (tempo + 1)$
---	---

Busca em Profundidade

- Legenda**
 -  Vértice desconhecido
 -  Vértice encontrado
 -  Vértice encontrado, com fecho positivo totalmente visitado
- d : marcador do instante que o vértice c foi descoberto;
- f : marcador do instante que o fecho transitivo do vértice c foi totalmente visitado (considerado então finalizado).

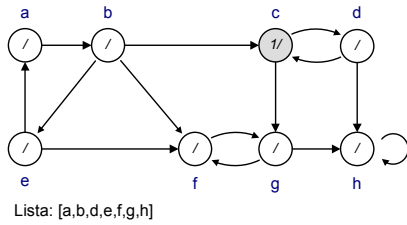
Busca em Profundidade

- Dado um Grafo, temos uma lista de todos os vértices:



Busca em Profundidade

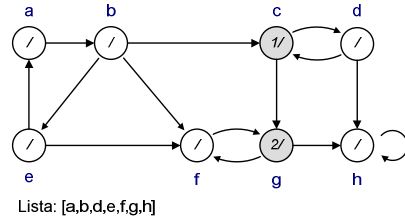
- Última ação: Vértice *c* é encontrado



Contador = 1

Busca em Profundidade

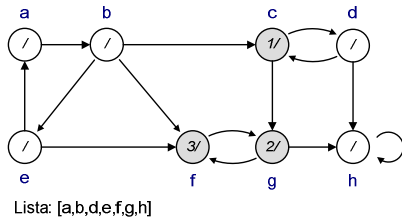
- Última ação: Vértice *g* é encontrado



Contador = 2

Busca em Profundidade

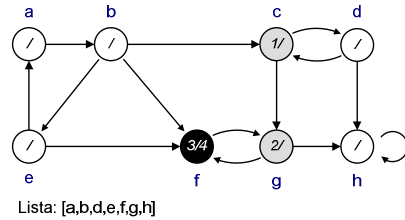
- Última ação: Vértice *f* é encontrado



Contador = 3

Busca em Profundidade

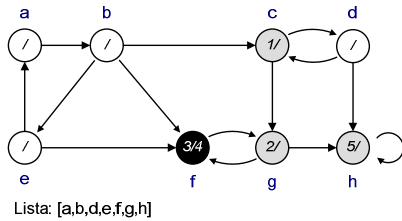
- Última ação: Vértice *f* é finalizado



Contador = 4

Busca em Profundidade

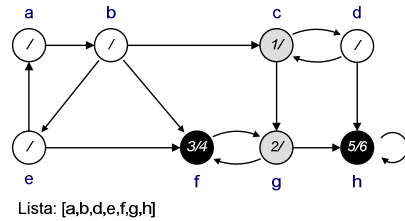
- Última ação: Vértice *h* é encontrado



Contador = 5

Busca em Profundidade

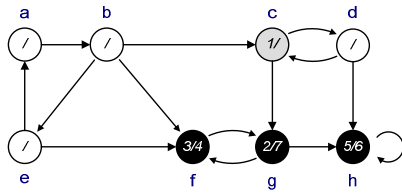
- Última ação: Vértice *h* é finalizado



Contador = 6

Busca em Profundidade

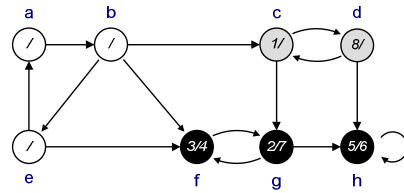
- Última ação: Vértice g é finalizado



Contador = 7

Busca em Profundidade

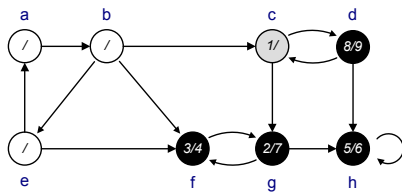
- Última ação: Vértice d é encontrado



Contador = 8

Busca em Profundidade

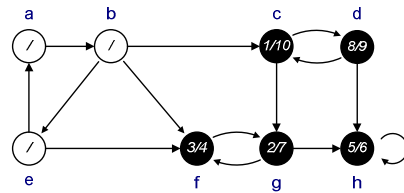
- Última ação: Vértice d é finalizado



Contador = 9

Busca em Profundidade

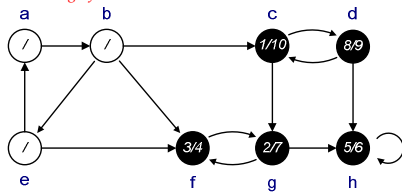
- Última ação: Vértice c é finalizado



Contador = 10

Busca em Profundidade

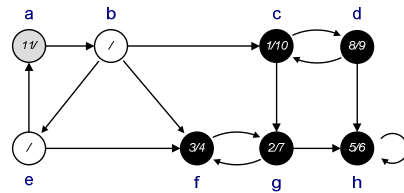
- Última ação: Capturando o próximo vértice da lista de vértices disponíveis no grafo



Contador = 10

Busca em Profundidade

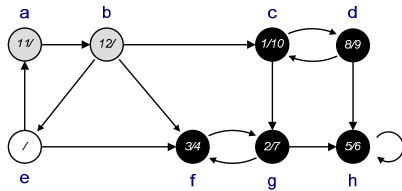
- Última ação: Vértice a é encontrado



Contador = 11

Busca em Profundidade

- Última ação: Vértice b é encontrado

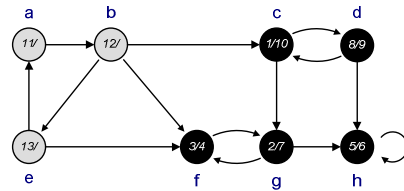


Lista: [b,d,e,f,g,h]

Contador = 12

Busca em Profundidade

- Última ação: Vértice e é encontrado

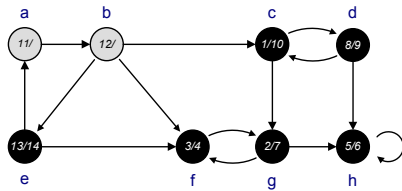


Lista: [b,d,e,f,g,h]

Contador = 13

Busca em Profundidade

- Última ação: Vértice e é finalizado

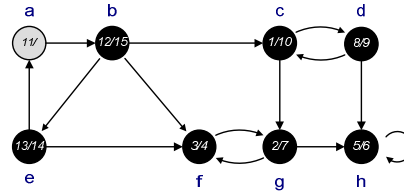


Lista: [b,d,e,f,g,h]

Contador = 14

Busca em Profundidade

- Última ação: Vértice b é finalizado

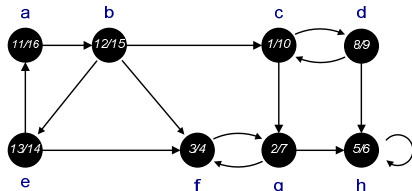


Lista: [b,d,e,f,g,h]

Contador = 15

Busca em Profundidade

- Última ação: Vértice a é finalizado

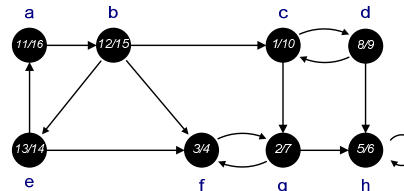


Lista: [b,d,e,f,g,h]

Contador = 16

Busca em Profundidade

- Última ação: Capturando os próximos vértices disponíveis. Será verificado que todos foram visitados pela busca!!!



Lista: [b,d,e,f,g,h]

Contador = 16

Busca em Profundidade

- Se você fosse implementar uma busca em profundidades otimizada, qual representação computacional escolheria?
 - Matriz de Adjacência;
 - Matriz de Incidência;
 - Lista de Adjacência;
- Por quê?

Exemplo

- No quadro...

Busca em Profundidade

<i>DFS(G)</i>	<i>DFS-VISIT(u)</i>
1 para cada vértice $u \leftarrow V[G]$	1 $cor[u] \leftarrow CINZA$
2 $cor[u] \leftarrow BRANCO$	2 $tempo \leftarrow tempo + 1$
3 $tempo \leftarrow 0$	3 $d[u] \leftarrow tempo$
4 para cada vértice $u \in V[G]$	4 para cada vértice $v \in Adj(u)$
5 se $cor[u] = BRANCO$	5 se $cor[v] = BRANCO$
6 $DFS-VISIT(u)$	6 $DFS-VISIT(v)$
	7 $cor[u] \leftarrow PRETO$
	8 $f[u] \leftarrow tempo \leftarrow (tempo + 1)$

Busca em Profundidade

- Bibliografia recomendada para esta aula:
 - “Algoritmos Teoria e Prática”, Tradução da 2ª edição. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. 2002, 936 pp. Editora Campus/Elsevier, ISBN 8535209263.

