

Aula 9: Busca em Grafos

Professor(a): Virgínia Fernandes Mota

<http://www.dcc.ufmg.br/~virginiaferm>

ALGORITMOS E ESTRUTURAS DE DADOS - SETOR DE INFORMÁTICA



- Busca em Largura (Breadth-First Search - BFS).
- Busca em Profundidade (Depth-First Search - DFS).

- A busca em largura é um dos algoritmos mais simples para exploração de um grafo.
- Dados um grafo $G = (V, E)$ e um vértice s , chamado de fonte, a busca em largura sistematicamente explora as arestas de G de maneira a visitar todos os vértices alcançáveis a partir de s .

- Esta busca é dita em largura porque ela expande a fronteira entre vértices conhecidos e desconhecidos de uma forma uniforme ao longo da fronteira.
- Ou seja, o algoritmo descobre todos os vértices com distância k de s antes de descobrir qualquer vértice de distância $k + 1$.

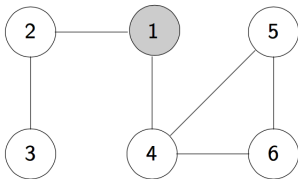
- Para controlar a busca, a BL (Busca em Largura) pinta cada vértice na cor branca, cinza ou preta.
- Todos os vértices iniciam com a cor branca e podem, mais tarde, se tornar cinza e depois preta.
 - Branca: não visitado
 - Cinza: visitado
 - Preta: visitado e seus nós adjacentes visitados

Algoritmo

- Quando um vértice é visitado pela primeira vez, sua cor é modificada de branco para cinza.
- Quando todos os vértices adjacentes a um vértice cinza são visitados, ele se torna preto.

Buscas em Grafos - Busca em Largura

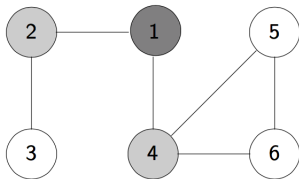
Início



	1	2	3	4	5	6
d	0	∞	∞	∞	∞	∞
π	\	\	\	\	\	\
c	g	w	w	w	w	w
Q	1					

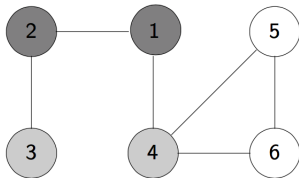
Buscas em Grafos - Busca em Largura

Explorando vértice 1



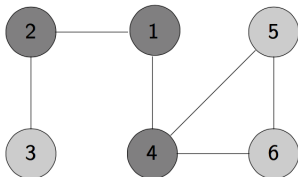
	1	2	3	4	5	6
d	0	1	∞	1	∞	∞
π	\	1	\	1	\	\
c	b	g	w	g	w	w
Q	2	4				

Explorando vértice 2



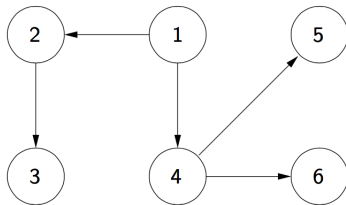
	1	2	3	4	5	6
d	0	1	2	1	∞	∞
π	0	1	2	1	\	\
c	b	b	g	g	w	w
Q	4	3				

Explorando vértice 3



	1	2	3	4	5	6
d	0	1	2	1	2	2
π	0	1	2	1	3	3
c	b	b	g	b	g	g
Q	3	5	6			

Árvore da busca em largura



Caminho mais curto

- Seja $\sigma(s, v)$ a distância do vértice v a partir do vértice s , sendo a distância o menor número de arestas em qualquer caminho em G com origem em s e destino para v .
- A busca em largura resolve o problema do caminho mais curto entre dois vértices. Nesse caso, o desempenho do algoritmo é melhor que os algoritmos de Dijkstra e Floyd.
- Outra situação onde a busca em largura pode ser usada é quando temos um grafo infinito. Nesse caso, a busca em profundidade pode entrar em um ramo sem saída.

- A estratégia aqui é explorar o grafo em profundidade.
- Na busca em profundidade, as arestas são exploradas a partir do vértice mais recentemente visitado.
- Da mesma forma que a busca em largura, sempre que um vértice v é descoberto durante a busca na lista de adjacência de um outro vértice já visitado u , a DFS memoriza este evento ao definir o predecessor de v , $\pi[v]$ como u .
- Diferentemente da BFS, cujo grafo predecessor forma uma árvore, o grafo predecessor de DFS pode ser composto de várias árvores (floresta).

- Grafo predecessor

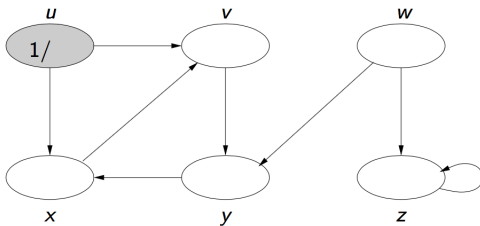
$$G_{\pi} = (V, E_{\pi})$$

$$E_{\pi} = \{(\pi[v], v) : v \in V \text{ e } \pi[v] \neq \text{NULL}\}$$

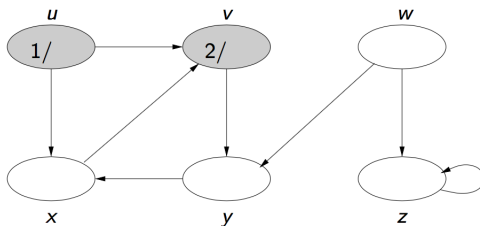
- Os vértices do grafo são coloridos durante a busca.
 - Branco: antes da busca.
 - Cinza: quando o vértice for visitado.
 - Preto: quando os vértices adjacentes foram visitados.

- Além de construir uma floresta, DFS marca cada vértice com um *timestamp*. Cada vértice tem dois *timestamp*.
 - $d[v] \rightarrow$ indica o instante em que v foi visitado (pintado com cinza).
 - $f[v] \rightarrow$ indica o instante em que a busca pelos vértices na lista de adjacência de v foi completada (pintado de preto).
- Usando timestamp 1, 2,..., verifica-se que:
 - $d[v], f[v] \in 1, \dots, 2|V|, \forall v \in V$
 - $d[v] < f[v], \forall v \in V$

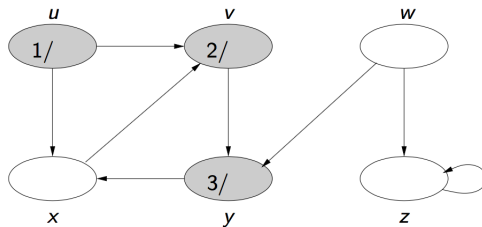
Buscas em Grafos - Busca em Profundidade



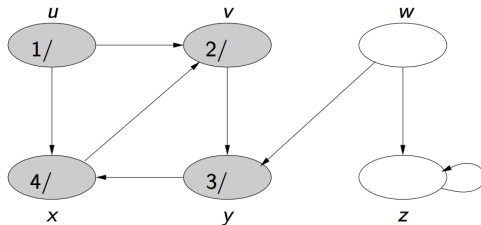
Buscas em Grafos - Busca em Profundidade



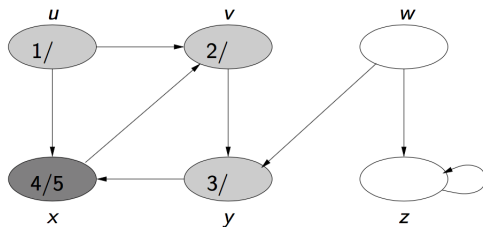
Buscas em Grafos - Busca em Profundidade



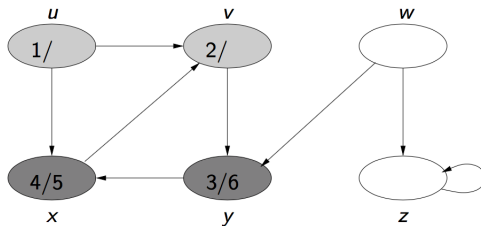
Buscas em Grafos - Busca em Profundidade



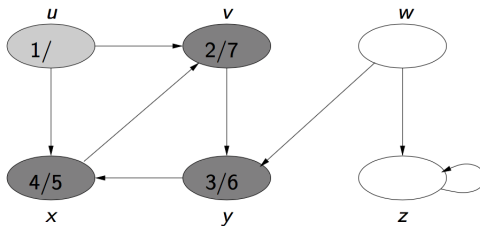
Buscas em Grafos - Busca em Profundidade



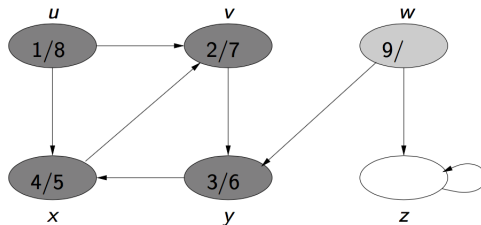
Buscas em Grafos - Busca em Profundidade



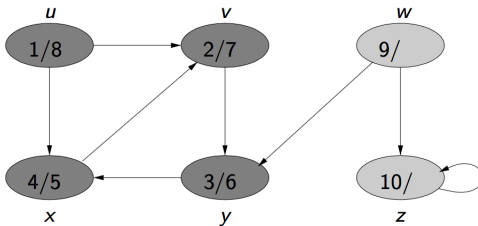
Buscas em Grafos - Busca em Profundidade



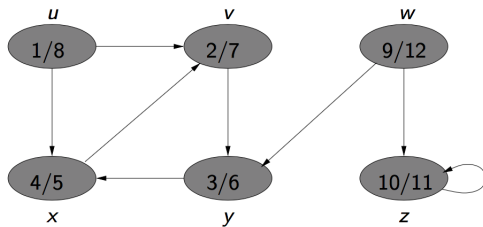
Buscas em Grafos - Busca em Profundidade



Buscas em Grafos - Busca em Profundidade



Buscas em Grafos - Busca em Profundidade



- A busca em profundidade tem aplicações em vários problemas:
- Teorema dos parênteses.
- Ordenação topológica.
- Identificação de componentes fortemente conexos.
- Coloração.

- 1 Implementar a Busca em Largura.
- 2 Implementar a Busca em Profundidade.

<https://www.cs.usfca.edu/~galles/visualization/BFS.html>

<https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>

Na próxima aula...

Busca, Ordenação e Hash