Busca em profundidade

Marco A L Barbosa
malbarbo.pro.br

Departamento de Informática
Universidade Estadual de Maringá

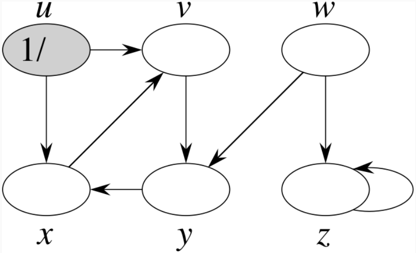
http://github.com/malbarbo/na-grafos

- Procurar “mais fundo” no grafo sempre que possível (*Depth-first search* - DFS em inglês)
- As arestas são exploradas a partir do vértices v mais recentemente descoberto que ainda tem arestas inexploradas saindo dele
- Quando todas as arestas de v são exploradas, a busca regressa para explorar as arestas que deixam o vértice a partir do qual v foi descoberto
- Este processo continua até que todos os vértices acessíveis a partir da origem tenham sidos descobertos
- Se restarem vértices não descobertos, a busca se repetirá para estes vértices

Introdução

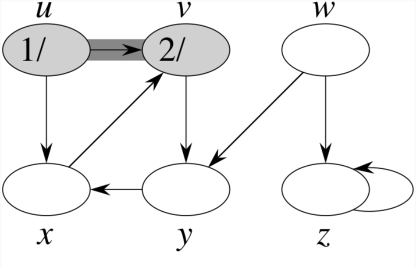
- Durante a execução do algoritmo, diversos atributos são definidos para os vértices
- Quando um vértice v é descoberto a partir de um vértice u, o campo predecessor $v.\pi = u$ é definido
- Cada vértice é inicialmente branco, o vértice é marcado de cinza quando é descoberto e marcado de preto quando é terminado (sua lista de adjacências é completamente examinada)
- Cada vértice tem dois carimbos de tempo $v.d$ (quando o vértice é descoberto) e $v.f$ (quando o vértice é terminado)

Exemplo de execução



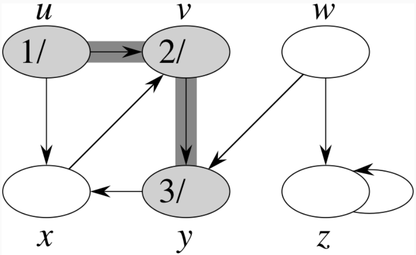
(a)

Exemplo de execução



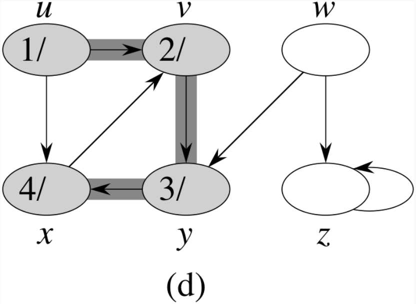
(b)

Exemplo de execução

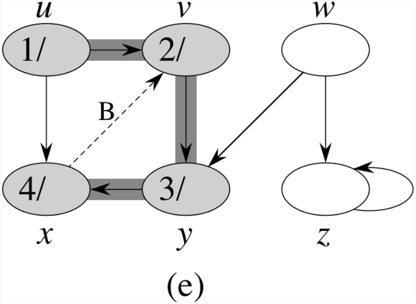


(c)

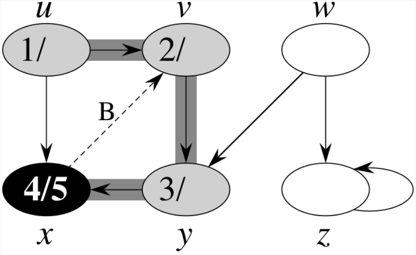
Exemplo de execução



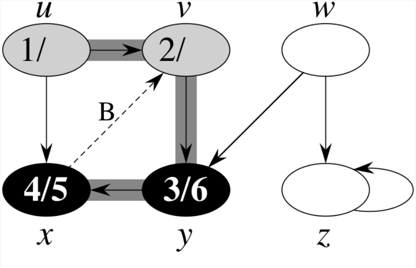
Exemplo de execução



Exemplo de execução

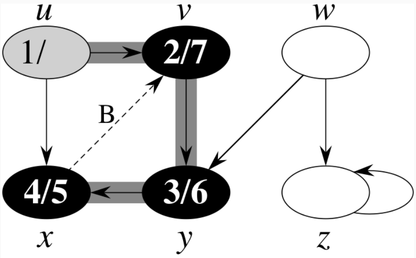


(f)



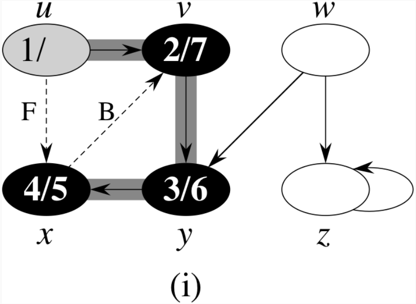
(g)

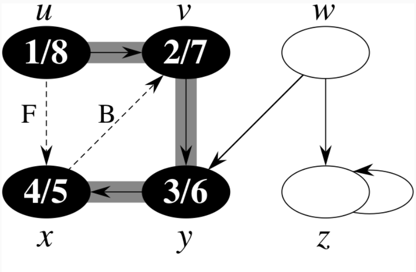
Exemplo de execução



(h)

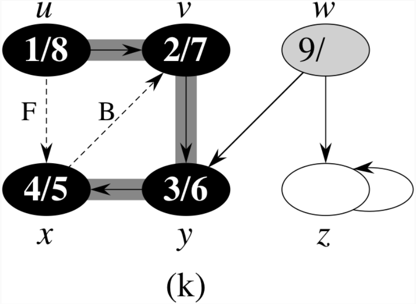
Exemplo de execução

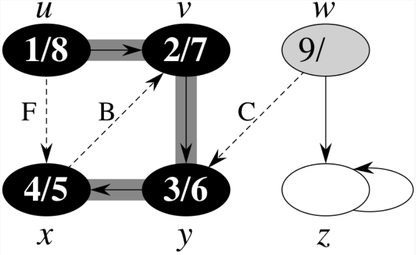




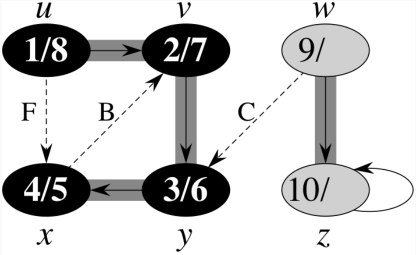
(j)

Exemplo de execução

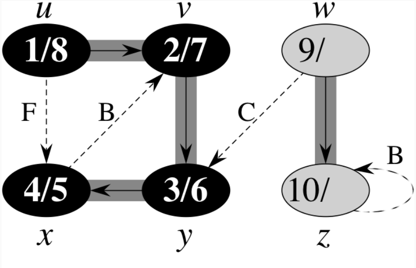




(l)

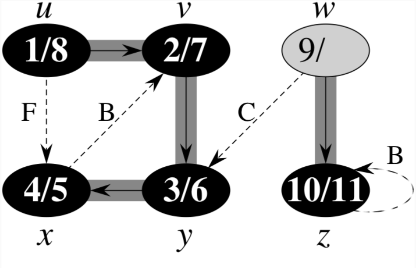


(m)

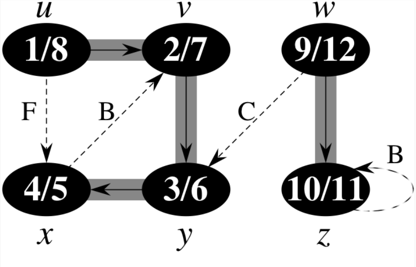


(n)

Exemplo de execução



(o)



(p)

Procedimento DFS

```
DFS(G)
1  for u ∈ G.V
2      u.cor = BRANCO
3      u.π = NIL
4  tempo = 0
5  for u ∈ G.V
6      if u.cor == BRANCO
7          DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
1  tempo = tempo + 1
2  u.d = tempo
3  u.cor = CINZA
4  for v ∈ G.Adj[u]
5      if v.cor == BRANCO
6          v.π = u
7          DFS-VISIT(G, v)
8  u.cor = PRETO
9  tempo = tempo + 1
10 u.f = tempo
```

Tempo de execução

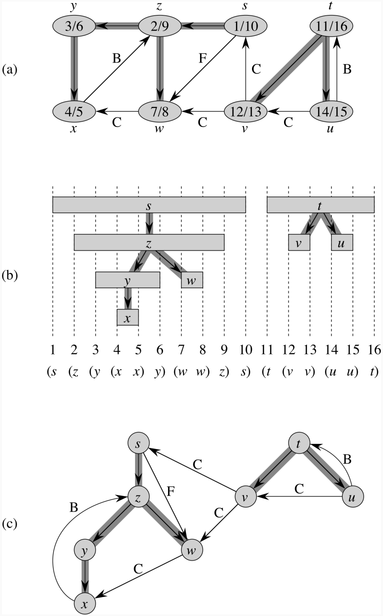
- Os laços nas linhas 1 a 3 e nas linhas 5 a 7 de DFS demoram tempo $\Theta(V)$, sem contar o tempo das chamadas a DFS-VISIT
- O procedimento DFS-VISIT é chamado exatamente uma vez para cada vértice, isto porque DFS-VISIT é chamado para os vértices brancos, e no início de DFS-VISIT o vértice é pintado de cinza
- Durante a execução de DFS-VISIT(v), o laço nas linhas 4 a 7 é executado $G.Adj[v]$ vezes, como $\sum_{v \in V} |G.Adj[v]| = \Theta(E)$, o custo total da execução das linhas 4 a 7 de DFS-VISIT é $\Theta(E)$
- Portanto, o tempo de execução do DFS é $\Theta(V + E)$

- O procedimento DFS constrói uma floresta da busca em profundidade (floresta DFS), contendo diversas árvores da busca em profundidade
- Para um grafo $G = (V, E)$, definimos o **subgrafo predecessor** de uma busca em profundidade de G como o grafo $G_\pi = (V, E_\pi)$ onde
    - $E_\pi = \{(v.\pi, v) : v \in V \text{ e } v.\pi \neq \text{NIL}\}$
- As arestas em E_π são **arestas da floresta**

Teorema 22.7 (Teorema do parênteses)
Para dois vértices quaisquer u e v, exatamente uma das três condições a seguir é verdadeira

- Os intervalos $[u.d, u.f]$ e $[v.d, v.f]$ são disjuntos e nem u e nem v são descendentes um do outro na floresta DFS
- O intervalo $[u.d, u.f]$ está contido inteiramente no intervalo $[v.d, v.f]$ e u é descendente de v em uma árvore DFS
- O intervalo $[v.d, v.f]$ está contido inteiramente no intervalo $[u.d, u.f]$ e v é descendente de u em uma árvore DFS

Prova feita em sala. Veja o livro para detalhes.



(a)
y z s t
3/6 2/9 1/10 11/16
B F C B
4/5 7/8 12/13 14/15
C C C
x w v u
(b)
s t z v u y w x
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
(s (z (y (x x) y) (w w) z) s) (t (v v) (u u) t)
(c)
s t
C B
z F v C u
B C
y w
C
x

Podemos definir quadro tipos de arestas em termos da floresta G_π

- **Arestas de árvore**, são as arestas em G_π. Uma aresta (u, v) é uma aresta de árvore se v foi descoberto primeiro pela exploração da aresta (u, v)
- **Arestas de retorno** são as arestas (u, v) que conectam um vértice u a um ancestral v em uma árvore DFS (consideramos laços como arestas de retorno)
- **Arestas diretas** são as arestas (u, v) que não são arestas da árvore e conectam o vértice u a um descendente v em uma árvore DFS
- **Arestas cruzadas** são todas as outras arestas

Quando uma aresta (u, v) é explorada, a cor do vértice v nos indica o tipo de aresta:

- BRANCO - aresta da árvore,
- CINZA - aresta de retorno, e
- PRETO - aresta direta ou cruzada

Classificação das arestas para grafos não orientados

Teorema 22.10
Na busca em profundidade de um grafo não orientado G, cada aresta de G ou é uma aresta de árvore ou uma aresta de retorno.

Prova feita em sala. Veja o livro para detalhes.

Referências

- Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.3.