

O estudo utilizando apenas este material **não é suficiente** para o entendimento do conteúdo. Recomendamos a leitura das referências no final deste material e a resolução (por parte do aluno) de todos os exercícios indicados.

Grafos

Busca em profundidade

Conteúdo

Introdução

Exemplo de execução

Procedimento dfs

Análise do tempo de execução do dfs

Floresta primeiro na profundidade

Propriedades

Exercícios

Referências

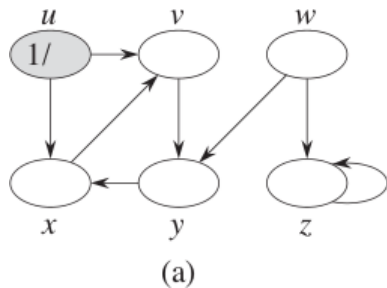
Introdução

- ▶ Procurar “mais fundo” no grafo sempre que possível
- ▶ 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 sido descobertos
- ▶ Se restarem vértices não descobertos, a busca se repetirá para estes vértices

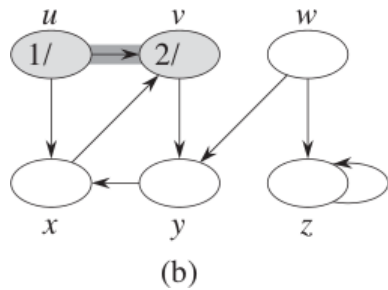
Introdução

- ▶ 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)

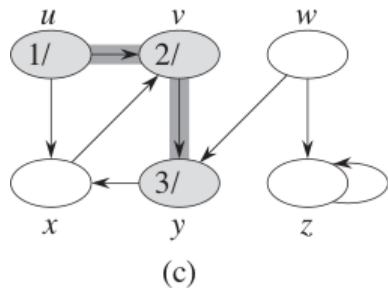
Exemplo de execução



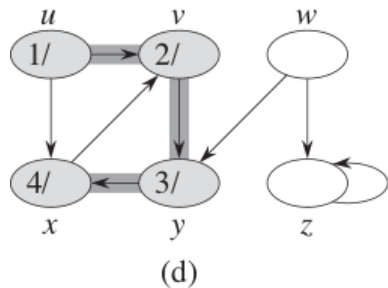
Exemplo de execução



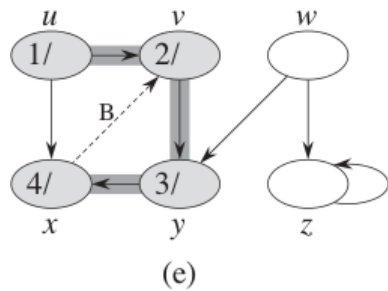
Exemplo de execução



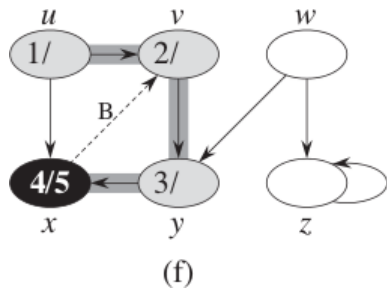
Exemplo de execução



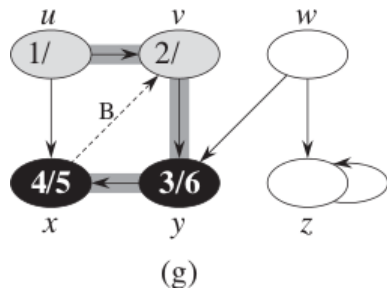
Exemplo de execução



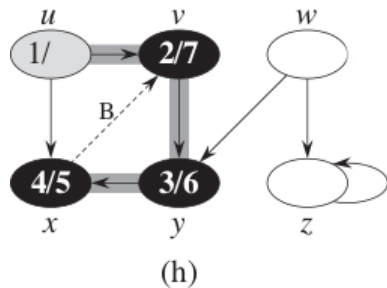
Exemplo de execução



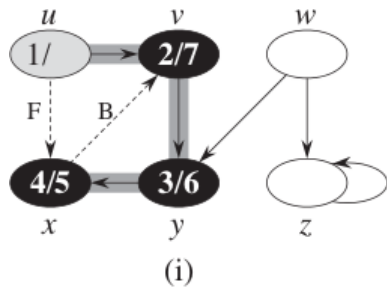
Exemplo de execução



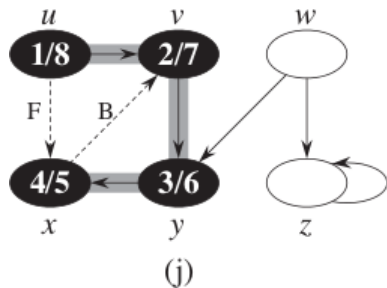
Exemplo de execução



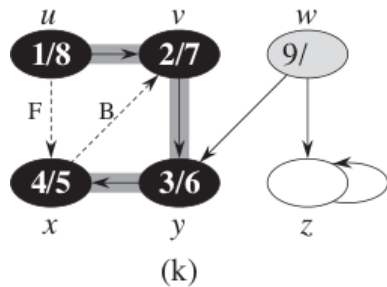
Exemplo de execução



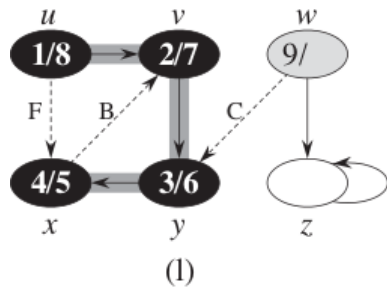
Exemplo de execução



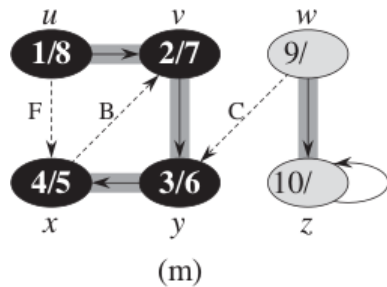
Exemplo de execução



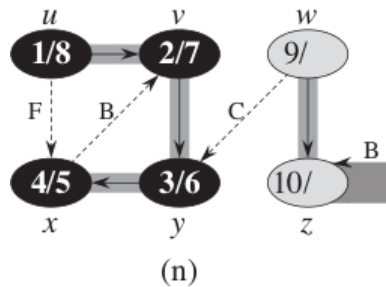
Exemplo de execução



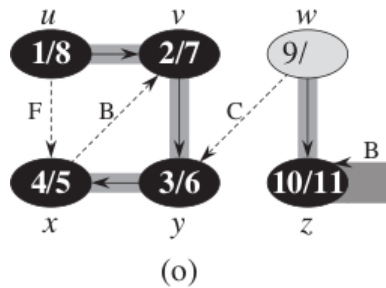
Exemplo de execução



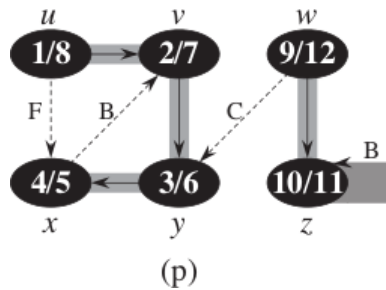
Exemplo de execução



Exemplo de execução



Exemplo de execução



Procedimento dfs

dfs(G)

```
1 for cada vértice u em G.V
2   u.cor = branco
3   u.pai = nil
4 tempo = 0
5 for cada vértice u em G.V
6   if u.cor == branco
7     dfs-visit(u)
```

dfs-visit(u)

```
1 tempo = tempo + 1
2 u.cor = cinza
3 u.d = tempo
4 for cada vértice v em u.adj
5   if v.cor == branco
6     v.pai = u
7     dfs-visit(v)
8 u.cor = preto
9 tempo = tempo + 1
10 u.f = tempo
```

Análise do tempo de execução do dfs

- ▶ Os loops 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
- ▶ Usamos a análise agregada
- ▶ 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 loop nas linhas 4 a 7 é executado $|v.adj|$ vezes, como $\sum_{v \in V} |v.adj| = \Theta(E)$, o custo total da execução das linhas 4 a 7 de dfs-visit é $\Theta(E)$
- ▶ Portanto, o tempo de execução do dfs é $\Theta(V + E)$

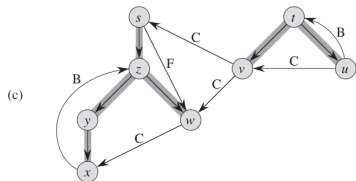
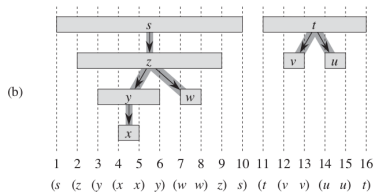
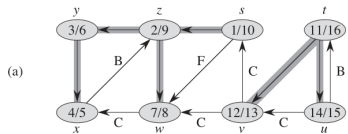
Floresta primeiro na profundidade

- ▶ DFS constrói uma floresta primeiro na profundidade, contendo diversas árvores primeiro na profundidade
- ▶ Para um grafo $G = (V, E)$, definimos o **subgrafo predecessor** de uma busca primeiro na 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 árvore**

Propriedades

- ▶ 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 primeiro na profundidade
 - ▶ O intervalo $[u.d, u.f]$ está contido inteiramente no intervalo $[v.d, v.f]$ e u é descendente de v em uma árvore primeiro na profundidade
 - ▶ O intervalo $[v.d, v.f]$ está contido inteiramente no intervalo $[u.d, u.f]$ e v é descendente de u em uma árvore primeiro na profundidade
 - ▶ Veja a prova no livro

Propiedades



Classificação das arestas

- ▶ Podemos definir quatro tipos de arestas em termos da floresta primeiro na profundidade G_π
 - ▶ **Arestas da árvore**, são as arestas na floresta primeiro na profundidade chamada G_π . Uma aresta (u, v) é uma aresta da 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 na árvore primeiro na profundidade
 - ▶ **Arestas para frente** são as arestas (u, v) que não são arestas da árvore e conectam o vértice u a um descendente v na árvore primeiro na profundidade
 - ▶ **Arestas cruzadas** são todas as outras arestas

Exercícios

- ▶ Os números dos exercícios referem-se a 3^o edição
- ▶ Entre parênteses está o número do exercício correspondente na 2^o edição
- ▶ Se não existe informação entre parênteses, significa que o exercício é o mesmo nas duas edições
- ▶ 22.3-1 a 22.3-3, 22.3-3 (não presente), 22.3-7 (22.3-6), 22.3-8 (22.3-7), 22.3-9 (22.3-8), 22.3-10 (22.3-9), 22.3-11 (22.3-10) e 22.3-12 (22.3-11)

Exercícios

22.3-1 Faça um diagrama 3 por 3 com linhas e colunas com identificações branco, cinza e preto. Em cada célula (i, j) , indique se, em qualquer instante durante uma busca em profundidade de um grafo orientado, pode existir uma aresta de um vértice de cor i até um vértice de cor j . Para cada aresta possível, indique quais tipos de arestas ela pode ser. Crie um segundo diagrama como esse para a busca em profundidade de um grafo não orientado.

Exercícios

22.3-2 Mostre como a busca em profundidade funciona sobre o grafo da figura 22.6. Suponha que o loop for das linhas 5 a 7 do procedimento `dfs` considere os vértices em ordem alfabética, e suponha que cada lista de adjacência esteja em ordem alfabética. Mostre os tempos de descoberta e término para cada vértice, e mostre também a classificação cada aresta.

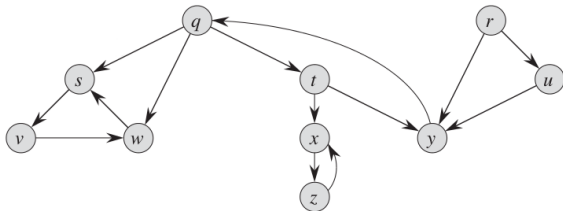


Figure 22.6 A directed graph for use in Exercises 22.3-2 and 22.5-2.

Exercícios

- 22.3-3 Mostre a estrutura de parênteses da busca em profundidade apresentada na figura 22.4
- 22.3-4 Mostre que usar apenas um bit para armazenar a cor de cada vértice é suficiente argumento que o procedimento `dfs` irá produzir o mesmo resultado se a linha 3 de `dfs-visit` for removida.
- 22.3-7 Reescreva o procedimento `dfs`, utilizando uma pilha para eliminar a recursão.
- 22.3-8 Forneça um contra-exemplo para a hipótese de que, se existe um caminho de u para v em um grafo orientado G , e se $u.d < v.d$ em uma busca em profundidade de G , então v é um descendente de u na floresta primeiro na profundidade produzida.

Exercícios

- 22.3-9 Forneça um contra-exemplo para a hipótese de que, se existe um caminho de u para v em um grafo orientado G , então qualquer busca em profundidade deve resultar em $v.d \leq u.f$.
- 22.3-10 Modifique o pseudocódigo para a busca em profundidade, de tal modo que ele imprima toda aresta no grafo orientado G , juntamente com seu tipo. Mostre quais modificações, se for o caso, devem ser feitas se G for não orientado.
- 22.3-11 Explique como um vértice u de um grafo orientado pode acabar em uma árvore primeiro na profundidade contendo apenas u , embora tenha tanto arestas de entrada quando de saída de G .

Exercícios

22.3-12 Mostre que uma busca em profundidade de um grafo não orientado G pode ser usada para identificar os componentes conexos de G , e que a floresta primeiro na profundidade contém tantas árvores quantos componentes conexos existem em G . Mais precisamente, mostre como modificar a busca em profundidade de modo que cada vértice v receba a atribuição de uma etiqueta inteira $v.cc$ entre 1 e k , onde k é o número de componentes conexos de G , de tal forma que $u.cc = v.cc$ se e somente se u e v estiverem no mesmo componente conexo.

Referências

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