Busca em profundidade

Algoritmos em Grafos

Marco A L Barbosa

Conteúdo

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.

Introdução

Introdução

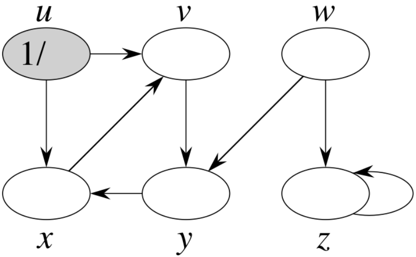
- Procurar "mais fundo" no grafo sempre que possível
- As arestas são exploradas a partir do vértices v mais recentemente descoberto que ainda tem arestas inexploradas saindo dele
- Quando todas as arestas de v são exploradas, a busca regressa para explorar as arestas que deixam o vértice a partir do qual v foi descoberto
- Este processo continua até que todos os vértices acessíveis a partir da origem tenham 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)

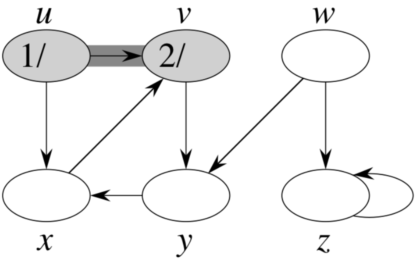
Exemplo de execução

Exemplo de execução



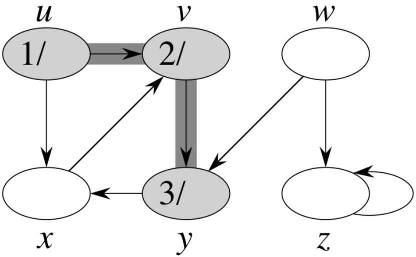
(a)

Exemplo de execução



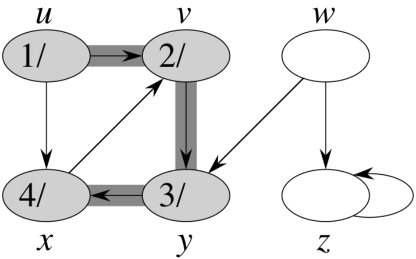
(b)

Exemplo de execução



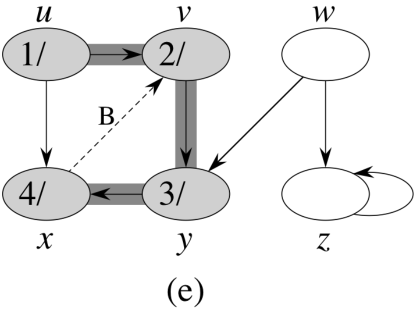
(c)

Exemplo de execução

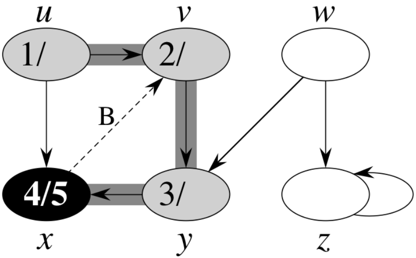


(d)

Exemplo de execução

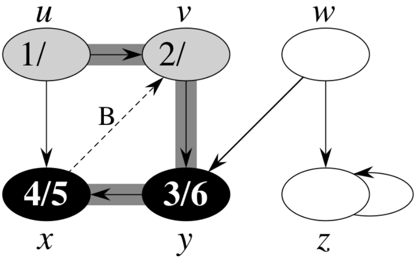


Exemplo de execução



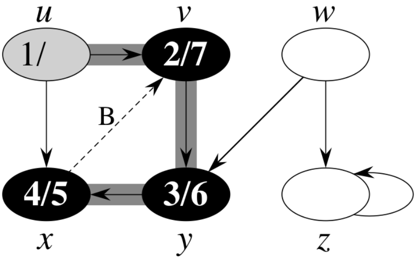
(f)

Exemplo de execução



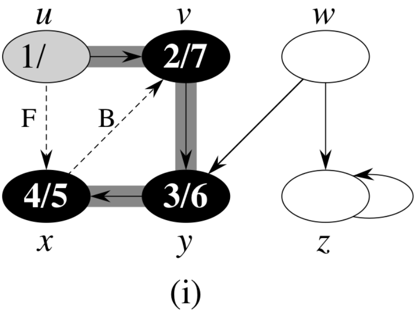
(g)

Exemplo de execução

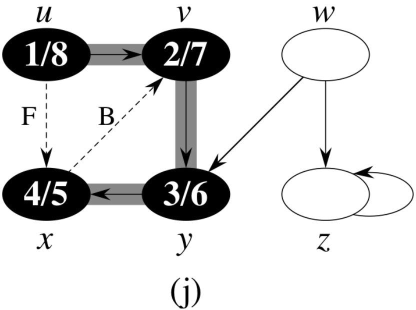


(h)

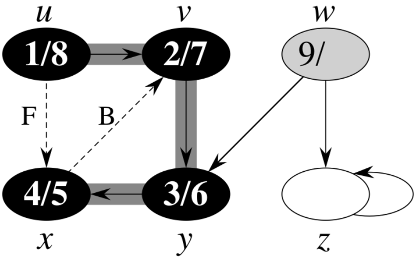
Exemplo de execução



Exemplo de execução

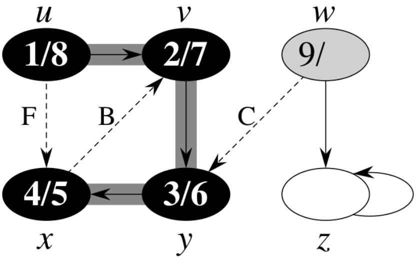


Exemplo de execução



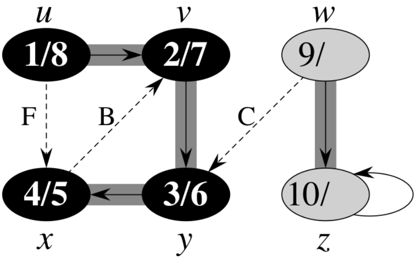
(k)

Exemplo de execução



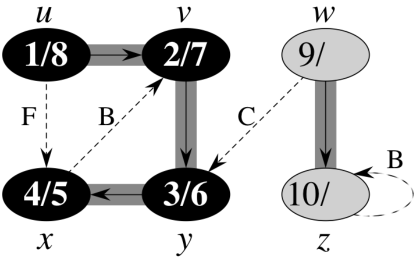
(l)

Exemplo de execução



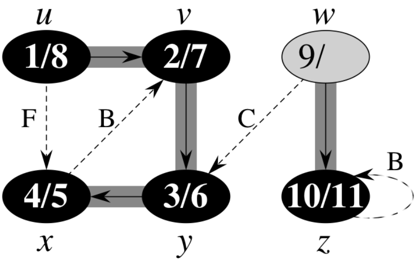
(m)

Exemplo de execução



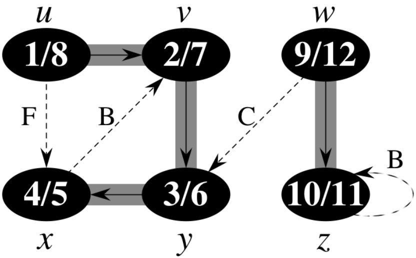
(n)

Exemplo de execução



(o)

Exemplo de execução



(p)

Procedimento `dfs`

Procedimento dfs

```
dfs(G)
for cada vértice u em G.V
  u.cor = branco
  u.pai = nil
tempo = 0
for cada vértice u em G.V
  if u.cor == branco
    dfs-visit(u)

dfs-visit(u)
tempo = tempo + 1
u.cor = cinza
u.d = tempo
for cada vértice v em u.adj
  if v.cor == branco
    v.pai = u
    dfs-visit(v)
u.cor = preto
tempo = tempo + 1
u.f = tempo
```

Análise do tempo de execução do `dfs`

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 laço 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 $\Theta(E)$

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 laço 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

Floresta primeiro na profundidade

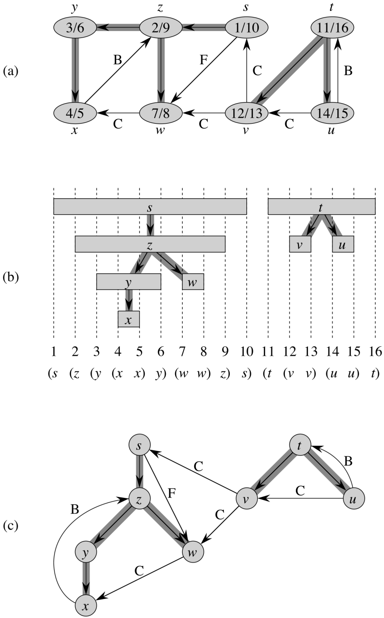
- O procedimento `dfs` constrói uma floresta primeiro na profundidade, contendo diversas árvores primeiro na profundidade
- Para um grafo $G = (V, E)$, definimos o **subgrafo predecessor** de uma busca primeiro na profundidade de G como o grafo $G_\pi = (V, E_\pi)$ onde
  - $E_\pi = \{(v.\pi, v) : v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

Propriedades

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

Propriedades



Classificação das arestas

- Podemos definir quadro 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

Referências

Referências

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