Projeto e Análise de Algoritmos

Algoritmos elementares de grafos – Parte 2

Diane Castonguay
diane@inf.ufg.br

Instituto de Informática
Universidade Federal de Goiás

Busca em profundidade

```
DFS(G)
1. para cada vértice u ∈ V[G] faça
2.         cor[u] ← BRANCO
3.          [u] ← NIL
4. fim-para
5. tempo ← 0
6. para cada vértice u ∈ V[G] faça
7.         se cor[u] = BRANCO
8.             então DFS-VISTIT(u)
9.         fim-se
10. fim-para
```

Subrotina DFS-VISIT

```
DFS-VISIT(u)
1. cor[u] ← CINZA
2. tempo ← tempo + 1
3. d[u] ← tempo
4. para cada vértice v ∈ Adj[u] faça
5.         se cor[v] = BRANCO
6.             então  [v] ← u
7.                    DFS-VISTIT(v)
8.         fim-se
9. fim-para
10. cor[u] ← PRETO
11. tempo ← tempo + 1
12. f[u] ← tempo
```

Teorema dos parêntheses

Em qualquer busca em profundidade de um grafo (orientado ou não orientado) G = (V, E), para dois vértices quaisquer u e v exactamente uma das três condições a seguir é válida:

- Os intervalos [d[u], f[u]] e [d[v], f[v]] são disjuntos, e nem u nem v é um descendente do outro na floresta primeiro na profundidade
-
- O intervalo [d[u], f[u]] está contido no intervalo [d[v], f[v]], e u é um descendente de v na árvore primeiro na profundidade
-
- O intervalo [d[v], f[v]] está contido no intervalo [d[u], f[u], e v é um descendente de u na árvore primeiro na profundidade

Classificação de arestas

- Arestas de árvore

são arestas na floresta primeiro na profundidade

- Arestas de retorno

são as arestas (u, v) que conectam um vértice u
a um ancestral v em uma árvore primeiro na profundidade

- Arestas direitas

são as arestas (u, v) que conectam um vértice u
a um descente v em uma árvore primeiro na profundidade

- Arestas cruzadas

as outras arestas

Ordenação topológica

/*
 * O grafo orientado G é acíclico.
 *As flechas vão da esquerda para a direita
 */

TOPOLOGICAL-SORT(G)
1. DFS(G) //calcular o tempo de término f[v]
2. A medida que f[v] é assinado, LIST-INSERT(L, f[v])
3. retorna L

Componentes fortemente conextados

/*
 * Um conjunto máximo de vértices é um componente
 * fortemente conectado se cada par de vértices são acessíveis
 * um a partir do outro.
 */

STRONGLY-CONNECTED-COMPONENTS(G)

1. DFS(G) //calcular o tempo de término f[v]
2. Calcular G^T
3. DFS(G^T) // no loop principal, considerar os vértices em
   // ordem decrescente de f[u], calculada na linha 1
4. cada árvore na floresta primeiro na profundidade formada na linha 3 é um componente fortemente conectado.