MO417 - Ata do exercício 22.4-3[1,2]

Enunciado:

Forneça um algoritmo que determine se um dado grafo não orientado $G = (V, E)$ contém um ciclo. Seu algoritmo deve ser executado no tempo $O(V)$, independente de $|E|$.

A solução aqui apresentada é de autoria de Renato Manzoni e baseia-se no algoritmo DFS (busca em profundidade).

Solução:

Em um grafo não orientado, um caminho $v0, v1, ..., vk$ forma um (simples) ciclo se $k \geq 3$, $v0 = vk$, e $v1, v2, ..., vk$ são distintos.[3]

As arestas de retorno de um grafo $G = (V, E)$ são arestas (u, v) tais que $(u, v) \in E$ e que conectam um vértice u ao seu antecessor v na árvore de busca em profundidade. Este tipo de aresta implica em um ciclo.

O algoritmo DFS e o procedimento DFS-VISITA foram modificados para retornar verdadeiro caso exista uma aresta de retorno no grafo e falso caso contrário. A marcação de tempo não é necessária para esse problema, por isto foi desconsiderada.

cor[u] : vetor que armazena a cor do vértice $u \in V$ e contém $|V|$ elementos. Cada vértice é inicialmente BRANCO, é CINZA quando é descoberto e é PRETO quando quando sua lista de adjacências foi totalmente examinada.

π[u] : vetor que armazena o antecessor de $u \in V$ e contém $|V|$ elementos. Se u não tem antecessor, então $\pi[u] = NIL$.

Adj[u] : lista que armazena todos os vértices $v \in V$ tais que existe a aresta $(u, v) \in E$, ou seja, Adj[u] é a lista de todos os vértices adjacentes a u em G.

ciclo : variável que indica se foi encontrado um ciclo até o momento

1 O exercício é do capítulo 22 de [1].

2 A ata foi redigida por Celina d' Ávila Samogin.

3 Apêndice B.4 Grafos de [1].

```
DFS-CICLO(G)
01  ciclo <- falso
02  para cada vertice u ∈  V[G] faca
03     cor[u] <- BRANCO
04     π[u] <- NIL
05  para cada vertice u ∈  V[G] faca
06     se cor[u] = BRANCO
07     entao ciclo <- (ciclo ou DFS-VISITA(u))
08  retorna ciclo
```

```
DFS-VISITA(u)
01  cor[u] <- CINZA
02  para cada v ∈  Adj[u] faca
03     se cor[v] = CINZA
04     entao retorna verdadeiro
05     senao se cor[v] = BRANCO
06           entao π[v] <- u
07                 DFS-VISITA(v)
08  cor[u] ← PRETO
09  retorna falso
```

Analisando as linhas de `02-04` (complexidade $\in \Theta(|V|)$) e as linhas de `05-07` (cuja complexidade $\in \Theta(|V|)$), o algoritmo `DFS-CICLO` $\in \Theta(|V|$. O procedimento `DFS-VISITA` é chamado uma vez para cada vértice $u \in V$.

No procedimento `DFS-VISITA(u)`, as linhas de `02-07` são executadas ou até encontrar um vértice já descoberto (ou seja, o primeiro ciclo) ou até percorrer todos os vértices adjacentes a `u`, após `|Adj[u]|` vezes.

Sabemos que:

$\Sigma|Adj[u]| \in \Theta(|E|)$

e pelo teorema B.2 (propriedades de árvores), se G = (V, E) é um grafo não dirigido:

G é acíclico e |E| = |V| -1 e

G é acíclico, mas se uma aresta for adicionada a E, o grafo resultante contém um ciclo.

Então o custo das linhas `02-07` de `DFS-VISITA` quando se percorre todos os vértices adjacentes a `u`:

é $\Theta(|E|) = \Theta(|V|-1) = \Theta(|V|)$

O tempo total do algoritmo `DFS-CICLO` é:

- $\Theta(|V|+|V|) \in \Theta(|V|)$, quando o grafo não tem ciclo ou quando é encontrado o primeiro ciclo do grafo

Referência:

[1] Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest e Clifford Stein. *Introduction to Algorithms. 2nd* edition.