

Caminhos e ciclos em grafos

Os conceitos de caminho e ciclo são essenciais no estudo de grafos. É preciso fazer distinções um tanto sutis entre caminhos, caminhos simples, passeios, ciclos, e ciclos simples.

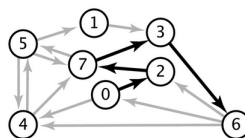
Sumário:

- Caminhos
- Ciclos

Caminhos

Um *passeio* (= *walk*) em um grafo é uma sequência de vértices dotada da seguinte propriedade: se v e w são vértices consecutivos na sequência então $v-w$ é um arco do grafo. (Note que o inverso de um passeio não é, em geral, um passeio.) Um *arco do passeio* é qualquer arco $v-w$ do grafo tal que w é o sucessor de v no passeio. Um passeio é *fechado* (= *closed*) se tem pelo menos dois arcos e seu primeiro vértice coincide com o último.

Um *caminho* (= *path*) em um grafo é um passeio sem arcos repetidos, ou seja, um passeio em que os arcos são todos diferentes entre si. Um caminho é *simples* se não tem vértices repetidos. Por exemplo, 0-2-7-3-6 é um caminho simples no grafo da figura.



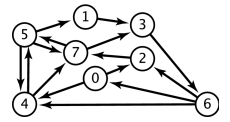
Todos os arcos de um caminho apontam na mesma direção — de um vértice para o seu sucessor. Há quem goste de enfatizar esse fato dizendo *caminho dirigido* em vez de *caminho*.

A *origem* de um caminho é o seu primeiro vértice. O *término* é o seu último vértice. Se um caminho tem origem s e término t , dizemos que *vai de s a t* .

O *comprimento* (= *length*) de um caminho é o número de arcos do caminho. Se um caminho tem n vértices, seu comprimento é pelo menos $n-1$; se o caminho é simples, seu comprimento é exatamente $n-1$.

Exemplo A. Considere, por exemplo, o grafo da figura e veja alguns caminhos nesse grafo:

0-2-7-3-6
 7-3
 7
 2-7-5-4-7-3
 5-1-3-6-4-5
 5-7-5



Os três primeiros caminhos são simples e têm comprimentos 4, 1 e 0 respectivamente. Os três últimos não são simples e têm comprimentos 5, 5 e 2 respectivamente. Para completar o exemplo, veja cinco sequências de vértices que *não* são caminhos:

1 3 6 2 7 3 6 4
 2 7 5 4 7 5 1
 6 3 7 2 0
 1 3 7
 1 7 3

As duas primeiras não são caminhos porque têm arcos repetidos.

Exemplo B. Considere o grafo cujos vértices são páginas da teia WWW e cujos arcos representam links (referências) de uma página a outra. Um passeio nesse grafo corresponde a uma pessoa que navega no espaço WWW seguindo os links.

Exemplo C. Considere o grafo cujos vértices são aeroportos e cujos arcos são voos comerciais entre aeroportos. Um caminho simples nesse grafo pode representar um roteiro de viagem com conexões.

Exercícios 1

1. *Todos os caminhos.* Faça uma lista de todos os caminhos simples com exatamente 4 vértices no grafo definido pelos arcos 7-3 1-4 7-8 0-5 5-2 3-8 2-9 0-6 4-9 2-6 6-4.
2. [Sedgewick 17.4] *Todos os caminhos.* Faça uma lista de todos os caminhos simples com exatamente 4 vértices no grafo não-dirigido definido pelas arestas 3-7 1-4 7-8 0-5 5-2 3-8 2-9 0-6 4-9 2-6 6-4.
3. *Verifica passeio.* Escreva uma função booleana `GRAPHcheckWalk()` que verifique se uma dada sequência `seq[0..k]` de vértices de um grafo é um passeio. Faça duas versões da função: uma supõe que o grafo é dado por sua matriz de adjacências e outra supõe que o grafo é dado por suas listas de adjacência.
4. *Passeio implica caminho.* Suponha que um grafo tem um passeio de um vértice x a um vértice y . Mostre que existe um caminho de x a y que usa um subconjunto dos arcos do passeio.
5. ★ *Caminho versus caminho simples.* Mostre que, para qualquer caminho P num grafo, alguma subsequência de P é um caminho simples com a mesma origem e o mesmo término de P . (Logo, existe um caminho de s a t se e somente se existe um caminho simples de s a t .)
6. *Verifica caminho simples.* Escreva uma função `GRAPHcheckSimplePath()` que verifique se uma sequência `seq[0..k]` de vértices de um grafo é um caminho simples. A função deve devolver -1 se a sequência não é um caminho, 1 se a sequência é um caminho simples, e 0 se a sequência é um caminho não simples. Faça duas versões da função: uma supõe que o grafo é dado por sua matriz de adjacências e outra supõe que o grafo é dado por listas de adjacência.

7. *Caminhos inversos.* Mostre que em qualquer [grafo não-dirigido](#) o [inverso](#) de um caminho também é um caminho.

Ciclos

Ciclos são estruturas muito importantes. São os ciclos que tornam grafos interessantes mas também complexos e difíceis de manipular.

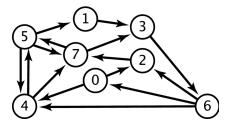
Um *ciclo* (= *cycle*) em um grafo é um caminho [fechado](#). (Portanto, todo ciclo tem [comprimento](#) maior que 1 e não tem arcos repetidos.) Dizemos que um arco $v-w$ *pertence* a um dado ciclo (ou que o ciclo *passa* pelo arco) se o vértice w é o sucessor de v no ciclo. Um ciclo é *simples* se não tem vértices repetidos exceto pelo último (que coincide com o primeiro).

É apropriado abusar um pouco do conceito de igualdade e tratar dois ciclos simples como *iguais* se eles têm o mesmo conjunto de arcos, ainda que tenham origens diferentes. Por exemplo, trataremos como iguais os ciclos 6-2-7-3-6, 2-7-3-6-2, 7-3-6-2-7 e 3-6-2-7-3.

Todos os arcos de um ciclo apontam no mesmo sentido — de um vértice do ciclo para o seu sucessor. Há quem goste de enfatizar esse fato dizendo *ciclo dirigido* no lugar de *ciclo*.

Exemplo D. Seguem alguns exemplos de ciclos no grafo da figura:

5-7-5
4-7-5-4
6-2-7-3-6
5-4-7-3-6-2-7-5



O primeiro tem comprimento 2, o segundo tem comprimento 3, e o terceiro tem comprimento 4. Esses três ciclos são simples. Já o último ciclo não é simples. Para completar o exemplo, veja dois passeios que não são ciclos:

6
5-4-7-5-1-3-6-2-7-5

Exercícios 2

- ★ Considere o grafo definido pelos arcos 0-1 1-2 2-0 2-3 3-1. A sequência 0-1-2-3-1-2-0 é um ciclo?
- Verifica ciclo.* Escreva uma função [booleana](#) que verifique se uma sequência $\text{seq}[0..k]$ de vértices de um grafo é um ciclo. Faça duas versões da função: uma supõe que o grafo é dado por sua [matriz de adjacências](#) e outra supõe que o grafo é dado por [listas de adjacência](#).
- Faça uma lista de todos os ciclos simples ([diferentes](#) entre si) no grafo definido pelos arcos 0-5 5-4 7-8 4-3 0-2 9-11 0-1 11-12 3-5 9-12 9-10 10-9 6-4 0-6.
- Faça uma lista de todos os [diferentes](#) ciclos simples no grafo definido pelo conjunto de arcos 0-4 1-0 2-1 2-3 3-0 3-4 4-1 4-2. Faça uma lista de todos os ciclos simples no grafo definido pelo conjunto de arcos 0-2 1-0 2-1 2-3 3-0 3-1.
- Faça uma lista de todos os ciclos simples no grafo definido pelo conjunto de arcos 0-2 1-0 2-1 2-3 3-0 3-1.

6. Faça uma lista de todos os ciclos simples no grafo [não-dirigido](#) definido pelo conjunto de arestas 0-1 1-2 2-3.
7. *Ciclos versus ciclos simples.* Seja a um arco de um grafo. Mostre que a pertence a um ciclo se e somente se a pertence a um ciclo simples.
8. *Ciclos em grafos sem sorvedouros.* Esboce um algoritmo para encontrar (e exibir) um ciclo simples num grafo sem [sorvedouros](#).

www.ime.usp.br/~pf/algoritmos_para_grafos/

Atualizado em 2017-04-08

© Paulo Feofiloff

IME-USP