1 - Introdução a Grafo

Um grafo (https://pt.wikipedia.org/wiki/Teoria_dos_grafos) é uma estrutura formada por nós e conexões, de tal forma que, as conexões representam relações entre os pares de nós conectados. Na prática essa estrutura pode ser utilizada, por exemplo, para representar conexões entre cidades. No exemplo da Figura 1, considere que os nós *a*, *b*, *c*, *d*, *e* sejam cidades e as conexões representam estradas que ligam essas cidades. Olhando o grafo é possível identificar os caminhos que podem ser percorridos para sair de uma cidade e chegar a outra.

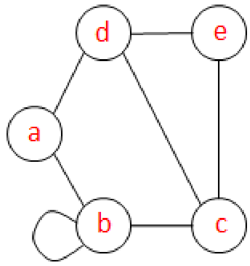

Figura 1 – Representação de um grafo.

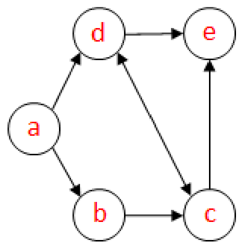

Figura 2 – Representação de um grafo orientado.

Termos importantes sobre grafos:

- Grafo também conhecido por grafo não orientado: é um conjunto de vértices e arestas.
  Uma aresta define um par não ordenado de vértices.
  Um grafo $G = (V, E)$ é composto de
  - V: conjunto de vértices (*vertex*, em inglês)
  - E: conjunto de arestas (*edges*, em inglês)

  Se $\alpha = (b, c)$ é uma aresta de G, dizemos que α liga os vértices *b* e *c*.
  No exemplo da Figura 1 tem-se:
  - $V = \{a, b, c, d, e\}$
  - $E = \{ (a,b), (b,b), (b,c), (c,d), (c,e), (a,d), (d,e)\}$
- Aresta direcionada ou também conhecida por arco: é a atribuição de um sentido a aresta, qualquer grafo construído desta forma é chamado de grafo orientado, assim como mostrado na Figura 2.
  Um arco $\alpha = (a,b)$ é considerado direcionado *de a para b*, onde *b* é chamado de cabeça e *a* é chamado de cauda do arco. Pode-se também dizer que, *b* é um sucessor direto de *a*. De forma indutiva pode-se dizer que, se um *caminho* composto por um ou mais arcos sucessivos leva *de a para e*, então *e* é um sucessor de *a* e *a* é um predecessor de *e*.
  O arco (b,a) é chamado de arco (a,b) invertido;
- Grafo orientado (*direct graph*, em inglês), também conhecido por digrafo, grafo dirigido e grafo direcionado: é um conjunto de vértices e arestas direcionadas (arcos), assim como mostrado na Figura 2. No grafo orientado os vértices só podem ser percorridos na direção das arestas direcionadas, pois não existe conexão na direção oposta. No exemplo da Figura 2 existe uma conexão (a,b) mas não existe uma conexão (b,a);
- Vértices adjacentes: são vértices conectados por uma aresta;
- Vizinhos: os vizinhos de um vértice é o conjunto de todos os vértices adjacentes a ele. Na Figura 1 os vértices *b* e *d* são vizinhos do vértice *a*;
- Relação de incidência: se uma aresta (a,b) conecta os vértices *a* e *b*, dizemos que *a* é incidente a *b*.
- Grau de um vértice: é o número de vértices adjacentes. A soma dos graus de todos os vértices do grafo é 2 vezes o número de arestas, no exemplo da Figura 1;

- Percurso: corresponde a uma sequência, finita e não vazia, de vértices do grafo, na qual (v_0, v_1, ..., v_{n-1}, v_n) é tal que, para todo $0 \leq i \leq n-1$, v_i e v_{i+1} são vértices adjacentes. Os vértices v_0 e v_n são chamados, respectivamente, de origem e fim do percurso, enquanto v_1, v_2, ..., v_{n-1} são os vértices internos ao caminho. n é o comprimento do percurso;
- Caminho: é um percurso que liga dois vértices do grafo;
- Laço: uma aresta que liga o vértice a ele mesmo, na Figura 1 existe um laço no vértice *b*;
- Grafo orientado simétrico: um grafo *G* é chamado de simétrico se, para cada arco pertencente a *G*, o arco invertido correspondente também pertence a *G*. Um grafo orientado simétrico sem laços é equivalente a um grafo não orientado com os pares de arcos invertidos substituído por arestas, assim o número de arestas é igual a metade do número de arcos;
- Grafo ponderado: é um grafo com pesos atribuídos as arestas.

Existem várias formas de representar computacionalmente um grafo, porém as mais usadas são baseadas em matrizes e listas:

- Lista de Adjacência: associa a cada vértice do grafo uma lista de todos os outros vértices com os quais ele tem alguma aresta;
- Matriz de Adjacência: uma matriz onde as linhas e colunas representam vértices, uma célula da matriz assume valor 1 quando os dois vértices são adjacentes e 0 caso contrário;
- Matriz de Incidência: uma matriz com suas linhas representando vértices e suas colunas representando as arestas. Cada célula assume um valor 0 ou 1 indicando se existe aresta incidente no vértice.

2 - Lista de Adjacência

Em teoria dos grafos, uma lista de adjacência (https://pt.wikipedia.org/wiki/Lista_de_adjacência) é a representação de todas as arestas ou arcos de um grafo em uma estrutura do tipo lista. Se o grafo é não orientado, cada entrada é um conjunto de dois nós contendo as duas extremidades da aresta correspondente; se ele for orientado, cada entrada é uma *tupla* de dois nós, um indicando o nó de origem e o outro o nó destino do arco correspondente. Normalmente, as listas de adjacência são desordenadas.

A Figura 3 e Figura 4 mostram representações das listas de adjacências dos grafos da Figura 1 e Figura 2, respectivamente. Uma lista de adjacência pode ser construída somente com arrays ou listas encadeadas, mas neste exemplo optou-se por criar um array de listas, onde cada elemento do array é um vértice que aponta para uma lista das suas arestas.

A principal alternativa para a lista de adjacência é a matriz de adjacência. Para um grafo esparso uma representação de lista de adjacências do grafo ocupa menos espaço que a matriz, porque ele não usa nenhum espaço para representar as arestas ausentes.

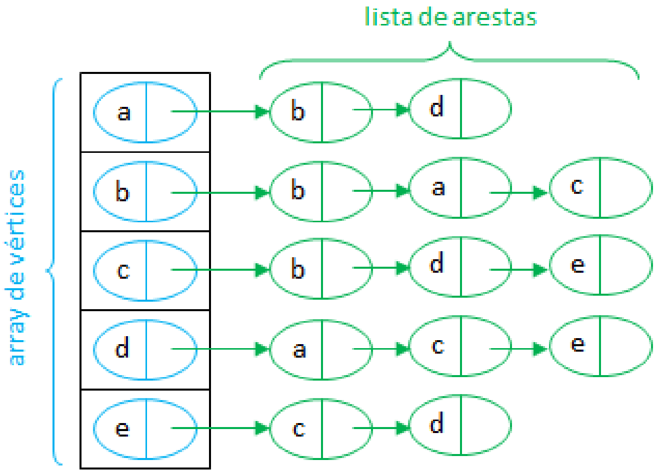

Figura 3 – Lista de Adjacência do grafo da Figura 1.

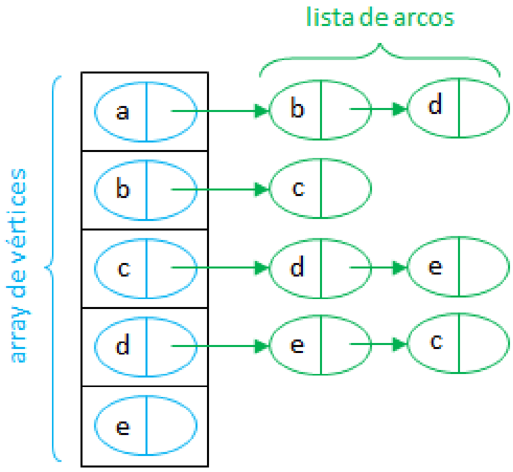

Figura 4 – Lista de Adjacência do grafo orientado da Figura 2.

3 - Matriz de Adjacência

Uma matriz de adjacência (https://pt.wikipedia.org/wiki/Matriz_de_adjacência) representa computacionalmente um grafo através de uma matriz bidimensional, onde cada dimensão possui os vértices.

A definição dos valores da matriz varia de acordo com as propriedades do grafo a ser representado, porém de forma geral cada célula guarda informações sobre como os vértices estão relacionados.

Para representar um grafo sem pesos nas arestas, basta que as células da matriz tenham:

- 1 se os vértices representados na linha e coluna da célula são adjacentes;
- 0 caso contrário.

Se as arestas do grafo tiverem pesos, use o peso da aresta ao invés de usar o valor 1.

A Figura 5 e Figura 6 mostram representações das matrizes de adjacências dos grafos da Figura 1 e Figura 2, respectivamente.

Em grafos não direcionados, as matrizes de adjacência são simétricas ao longo da diagonal principal - isto é, a entrada c_{ij} é igual à entrada c_{ji}, já as matrizes de adjacência de grafos orientados não são simétricas.

	a	b	c	d	e
a	0	1	0	1	0
b	1	1	1	0	0
c	0	1	0	1	1
d	1	0	1	0	1
e	0	0	1	1	0

Figura 5 – Matriz de Adjacência do grafo da Figura 1.

para

de	a	b	c	d	e
a	0	1	0	1	0
b	0	0	1	0	0
c	0	0	0	1	1
d	0	0	1	0	1
e	0	0	0	0	0

Figura 6 – Matriz de Adjacência do grafo orientado da Figura 2.

4 - Matriz de Incidência

Uma matriz de incidência (https://pt.wikipedia.org/wiki/Matriz_de_incidência) representa computacionalmente um grafo através de uma matriz bidimensional, onde uma das dimensões são vértices e a outra são arestas.

Dado um grafo *G* com *n* vértices e *m* arestas, podemos representá-lo em uma matriz *M* de *n* x *m*. A definição precisa das entradas da matriz varia de acordo com as propriedades do grafo que se deseja representar, porém de forma geral guarda informações sobre como os vértices se relacionam com cada aresta (isto é, informações sobre a incidência de uma aresta em um vértice).

Para representar um grafo sem pesos nas arestas e não direcionado, basta que as entradas da matriz *M* contenham 1 se a aresta incide no vértice, 2 caso seja um laço (incide duas vezes) e 0 caso a aresta não incida no vértice. Veja como exemplo a Figura 7 e Figura 8.

arestas

vértices	*ab*	*ad*	*bb*	*bc*	*cd*	*ce*	*de*
a	1	1	0	0	0	0	0
b	1	0	1	1	0	0	0
c	0	0	0	1	1	1	0
d	0	1	0	0	1	0	1
e	0	0	0	0	0	1	1

Figura 7 – Matriz de Incidência do grafo da Figura 1.

arcos

vértices	*ab*	*ad*	*bc*	*cd*	*ce*	*dc*	*de*
a	-1	-1	0	0	0	0	0
b	1	0	-1	0	0	0	0
c	0	0	1	-1	-1	1	0
d	0	1	0	1	0	-1	-1
e	0	0	0	0	1	0	1

Figura 8 – Matriz de Incidência do grafo orientado da Figura 2.

5 - Exercícios

1 – Programar uma matriz de adjacência para representar o grafo da Figura 1. Use a classe da Figura 9.

2 – Programar uma matriz de incidência para representar o grafo da Figura 1. Use a classe da Figura 9.

3 – Programar uma lista de adjacência para representar o grafo da Figura 1.

Matriz
- m : int[][]
+ Matriz(linha : int, coluna : int) + setCelula(nro : int, linha : int, coluna : int) : void + imprimir() : void

Figura 9 – Classe Matriz.