

# ◀ Introdução a Grafos ▶

**Kaio Christaldo**  
**Fabricio Matsunaga**

# O que é um Grafo?

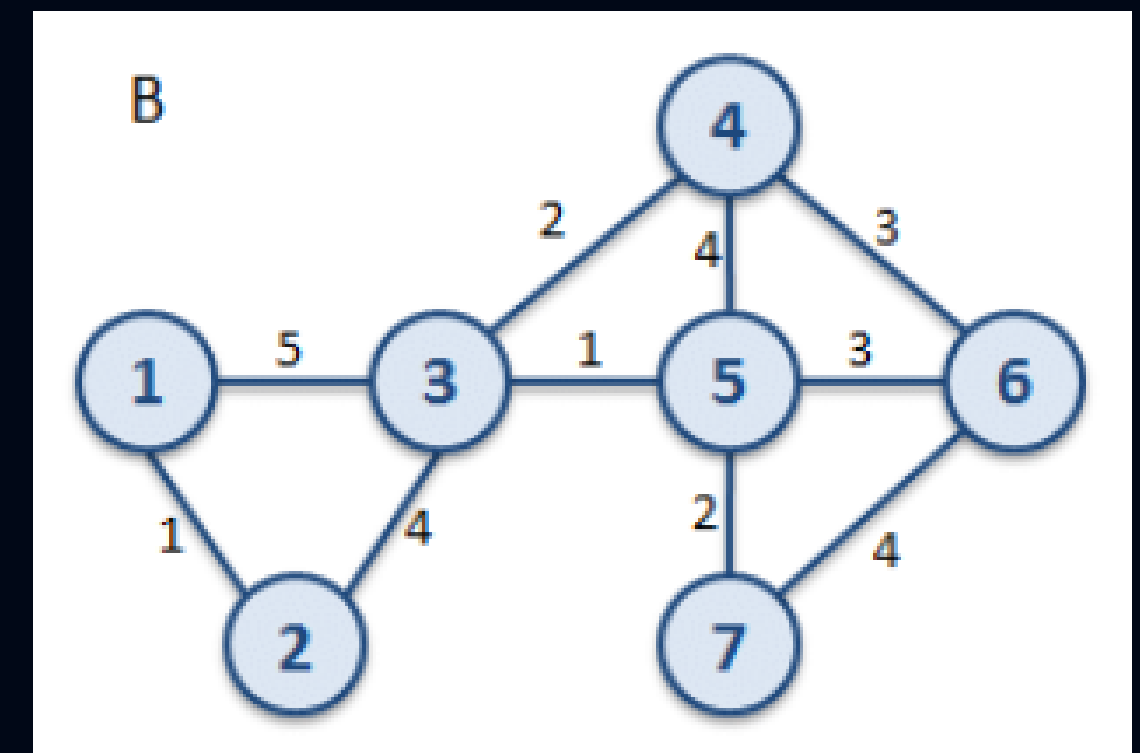
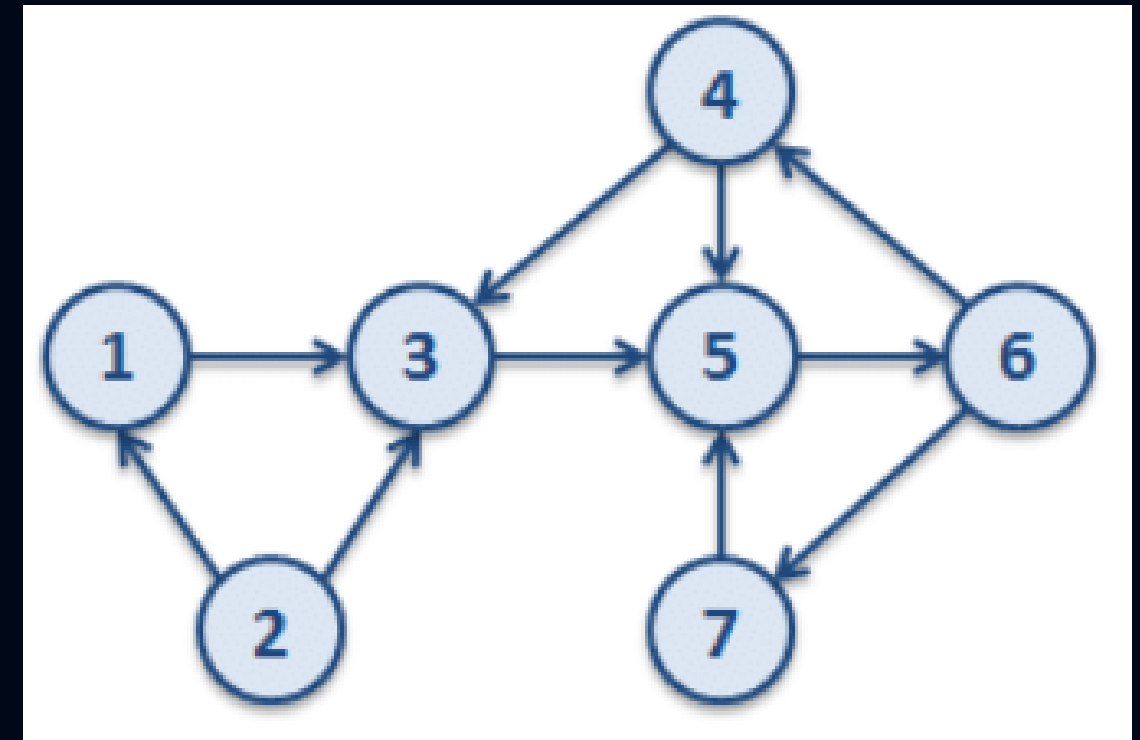
Um grafo é simplesmente um conjunto de pontos e linhas que conectam esses pontos.

- Um grafo é composto por:
  - Vértices (ou nós): São os "pontos". Eles representam os objetos de interesse (cidades, pessoas, páginas da web, etc.).
  - Arestas (ou arcos): São as "linhas". Elas representam as conexões ou relações entre os vértices (estradas, amizades, links, etc.).
- São representados em Maratonas de Programação como Matriz de Adjacência e Lista de Adjacência.

# Tipos de Grafos

## Grafos Direcionados e Não-Direcionados:

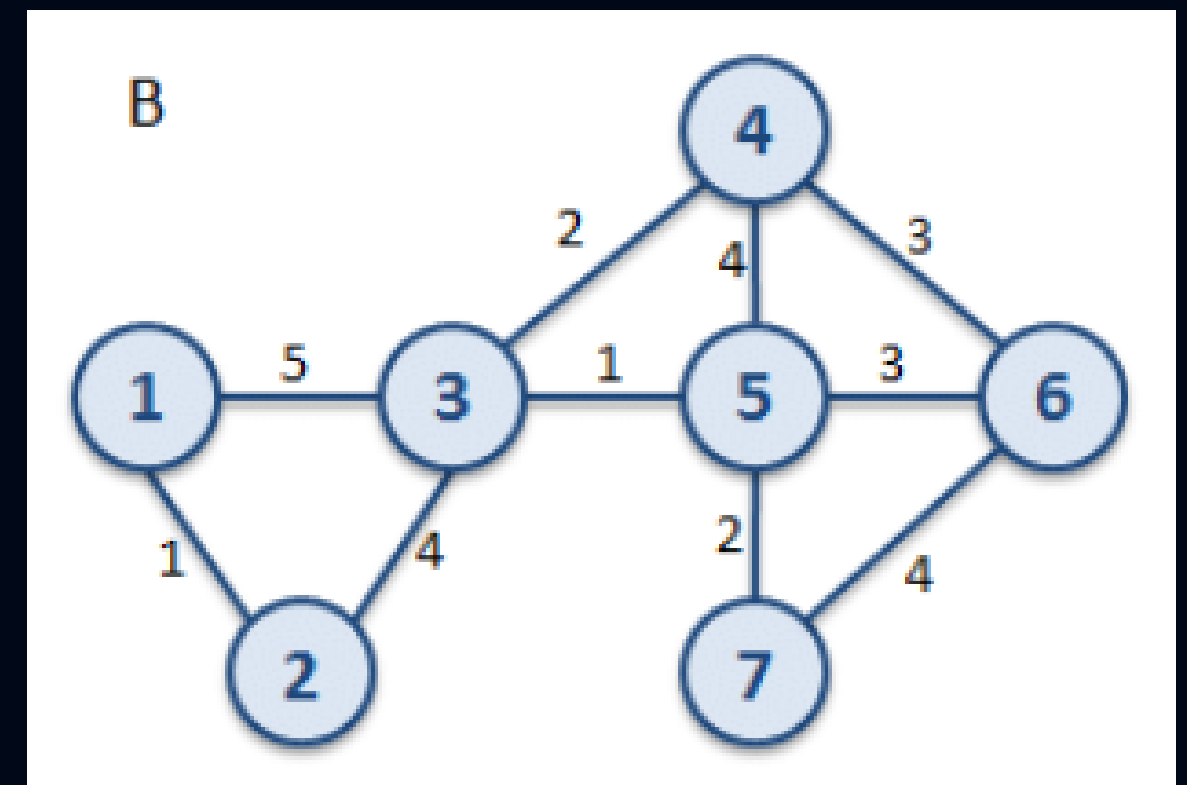
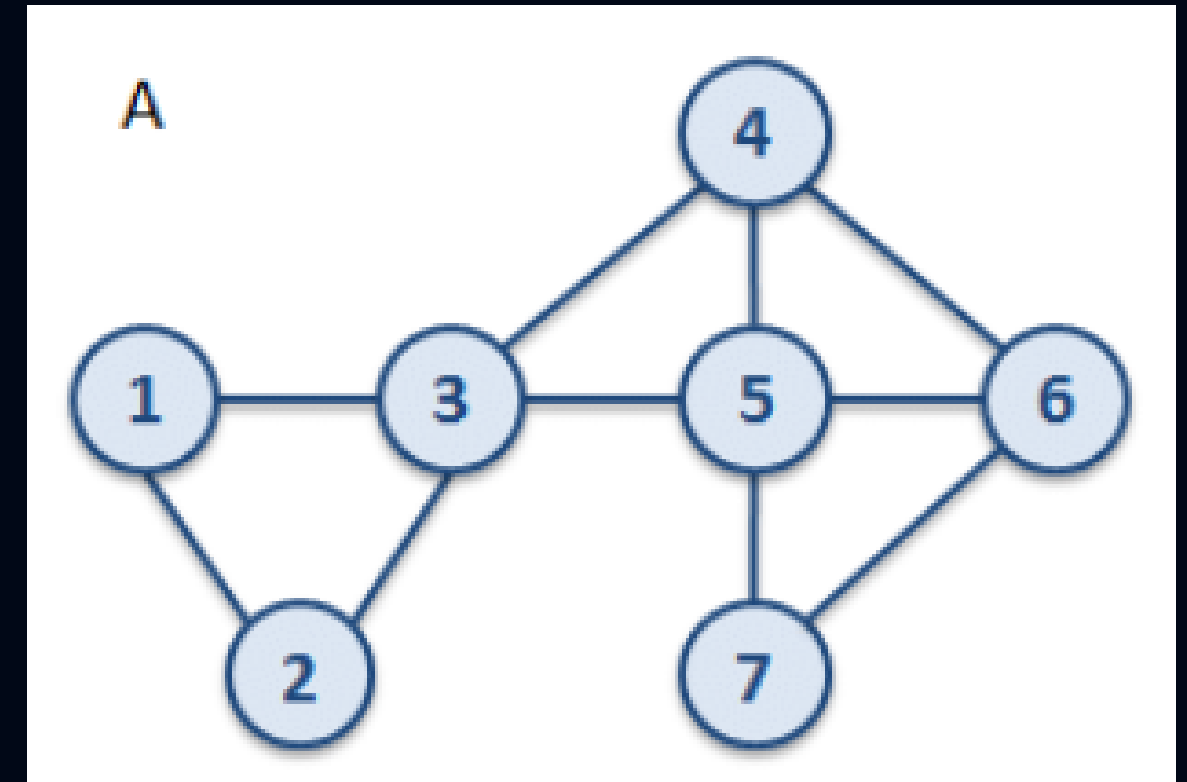
- **Não-Direcionado:** A conexão é mútua. Se A é amigo de B, B também é amigo de A. A aresta é uma via de mão dupla.
- **Direcionado:** A conexão tem um sentido. Uma rua de mão única é um bom exemplo. Seguir uma pessoa no Twitter não significa que ela te segue de volta. A aresta é uma "seta".



# Tipos de Grafos

## Grafos Com Peso e Sem Peso:

- **Sem Peso:** Todas as conexões são iguais. O que importa é se a conexão existe ou não.
- **Com Peso (Ponderado):** Cada conexão tem um "custo" ou "distância" associada. Por exemplo, a distância em quilômetros entre duas cidades ou o tempo de voo entre dois aeroportos.



# ◀Matriz de Adjacência▶

# **Apresentação Problema Motivador**

**Vou procurar um exercicios de Matriz de adjacencia**

**1161 – Soma  
de Fatoriais**

## ◀Matriz de Adjacência▶

Uma matriz de adjacência é uma das formas de se representar um grafo

Dado um grafo com  $n$  vértices, a matriz de adjacências é uma matriz booleana de  $n \times n$  casas. Se há alguma aresta ligando os vértices  $i, j$  (nessa ordem), então `matriz_adjacencias[i, j] == True`; caso contrário, seu valor é falso.

## ◀Matriz de Adjacência▶

Ela se encaixa em uma das estratégias de representação de grafos. Note que ela se aplica a grafos finitos e de vértices conhecidos. O que quero dizer com isso?, por acaso existem grafos infinitos?

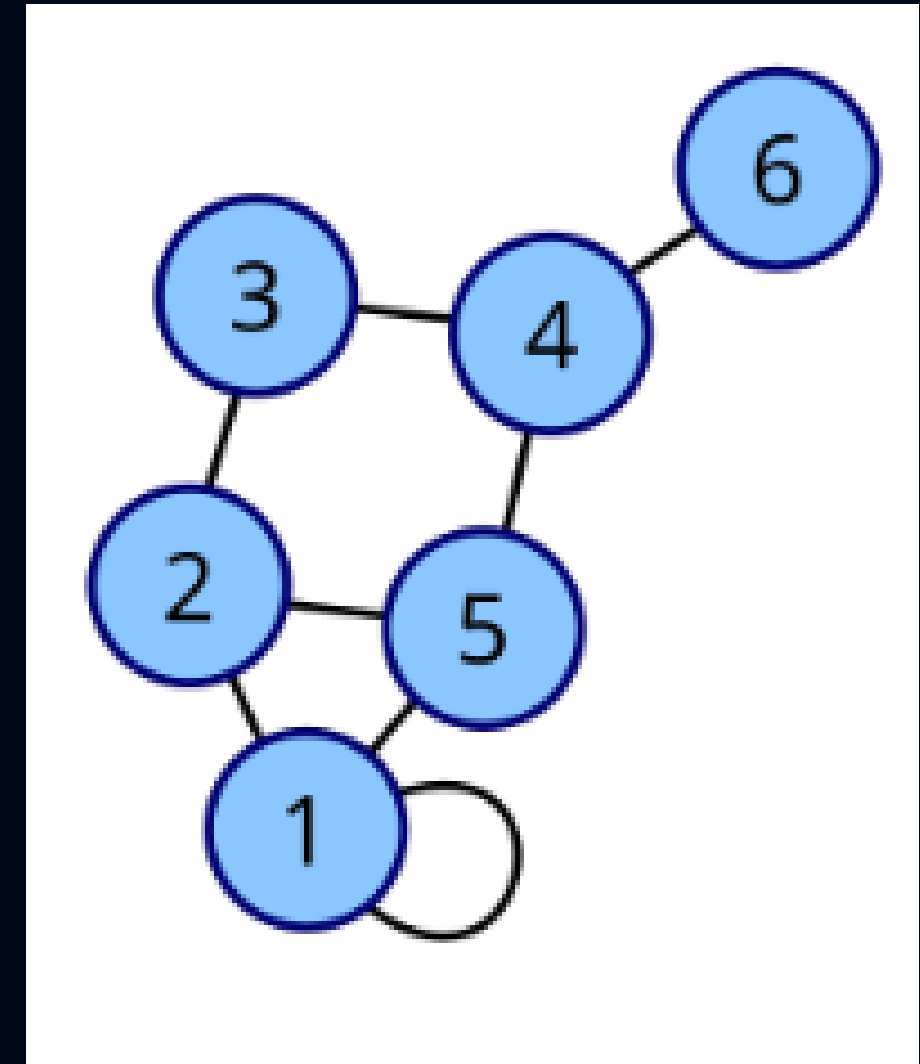
Sim, existem grafos infinitos. Eles são descritos proceduralmente.



## ◀Matriz de Adjacência▶

Por exemplo, a matriz de adjacência do grafo ao lado é

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$



## ◀Algoritmo Basico▶

### Exemplo em c++ usando vector



```
1 //Exemplo de matriz de adjacência
2 int n;
3 cin >> n;
4 vector<vector<int>> adj(n, vector<int>(n,0));
5 for (int i = 0; i < n; i++) {
6     for (int j = 0; j < n; j++) {
7         cin >> adj[i][j];
8     }
9 }
```

## ◀ Algoritmo Básico ▶

O grafo é orientado?

Se sim  $A[i][j] \neq A[j][i]$  senão  $A[i][j] == A[j][i]$

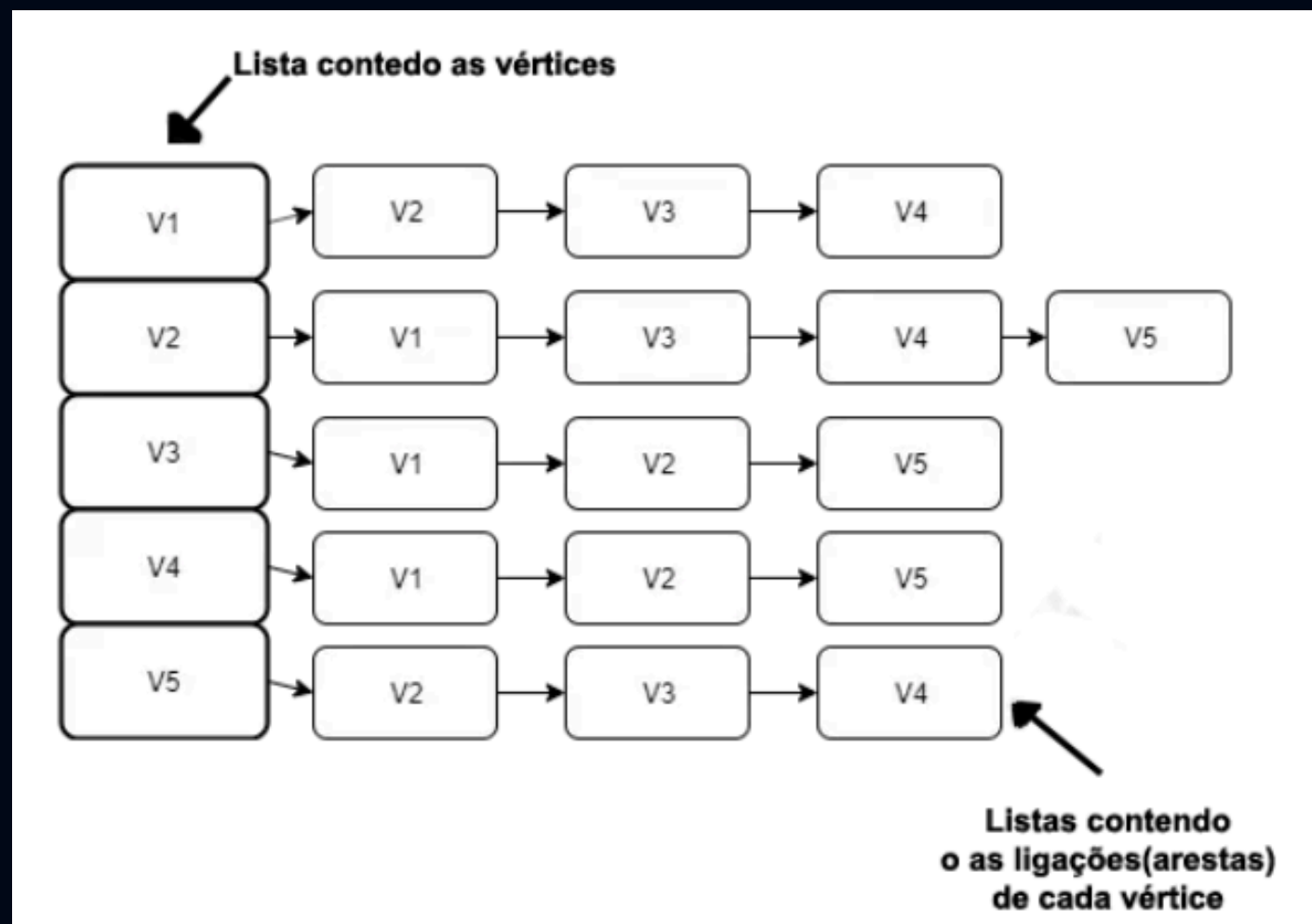
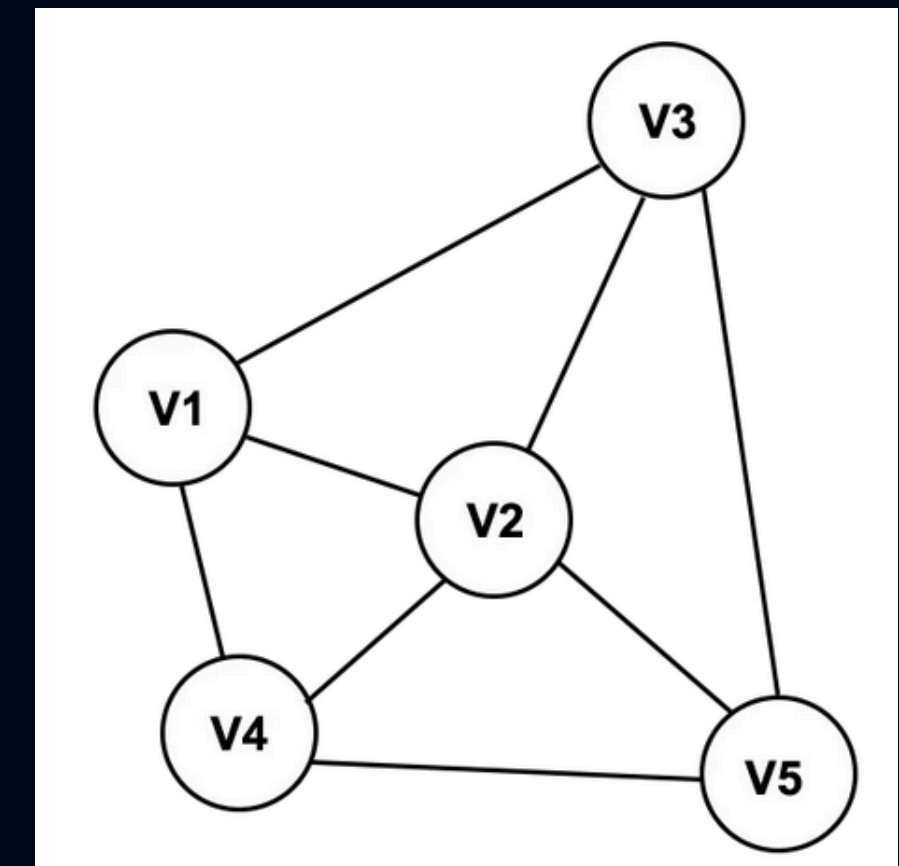
# Resolução do Problema Motivador

A resolução estará disponível no Drive. Tente resolver por conta própria e, se precisar, compare com a solução! 😊

# ◀Lista de Adjacência▶

# Lista de Adjacência

Consiste em uma lista qual cada elemento é um nó do grafo, logo a lista possui de tamanho o numero de vértices do grafo. Ela possui n lista ligadas, estas contem as conexões de cada arestas do vértice.



Computacionalmente, vamos representar isto como uma lista que em cada elemento (representando um vértice do grafo) existe um ponteiro para todas as arestas ligadas a ela

# Apresentação Problema Motivador

Você é o(a) novo(a) engenheiro(a) de software no time de desenvolvimento do "Metrópole Maps", o sistema de GPS mais avançado da cidade. Sua primeira tarefa é criar um módulo de consulta rápida para o sistema. Antes de calcular rotas complexas, o sistema precisa responder a uma pergunta muito básica, porém fundamental, milhares de vezes por segundo: "O ponto A é diretamente conectado ao ponto B por uma avenida? Se sim, qual é o comprimento dessa avenida?"

Sua tarefa é ler o mapa da cidade, que consiste em pontos de interesse (cruzamentos, monumentos, etc.) e as avenidas que os conectam, e depois processar uma série de consultas em tempo real.

Dado um mapa da cidade, onde os pontos de interesse são representados por números e as avenidas têm um comprimento em metros, você deve construir uma estrutura de dados que permita responder eficientemente a consultas sobre a adjacência e a distância direta entre dois pontos.

O grafo que representa o mapa é não-direcionado e ponderado.

# Apresentação Problema Motivador

- Leia a descrição do mapa de um arquivo de texto e armazene-o em uma estrutura de dados apropriada (uma lista de adjacência é uma ótima escolha, mas pense em como armazenar os pesos).
- Após carregar o mapa, leia uma série de consultas.
- Para cada consulta, verifique se dois pontos A e B são diretamente conectados.
- Se forem, imprima a distância em metros entre eles.
- Se não houver uma avenida direta entre eles, informe que não são adjacentes.

```
5 6
0 1 500
0 4 1200
1 2 350
1 3 800
1 4 250
3 4 400
3
0 1
2 4
4 3
```

```
Sim, por 500 metros
Nao sao adjacentes
Sim, por 400 metros
```