GRAFOS E OTIMIZAÇÃO

DCE770 - Heurísticas e Metaheurísticas

Atualizado em: 18 de agosto de 2025

Iago Carvalho

Departamento de Ciência da Computação

GRAFOS

Diversos problemas computacionais podem ser representados como grafos

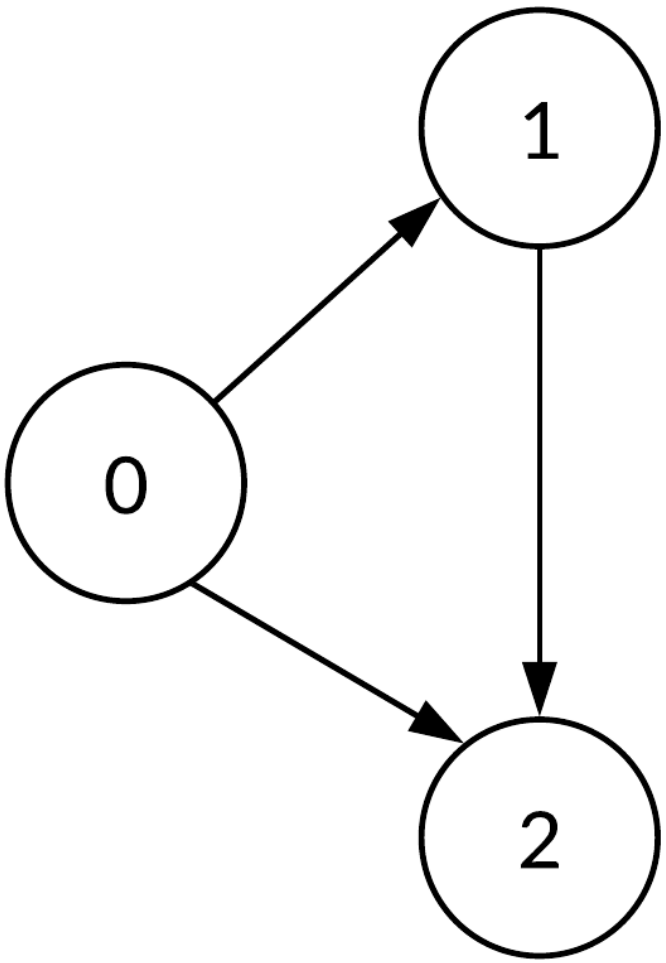
- Uma estrutura de dados especial
- Representação de uma rede
- Talvez seja a estrutura mais útil em toda a Ciência da Computação

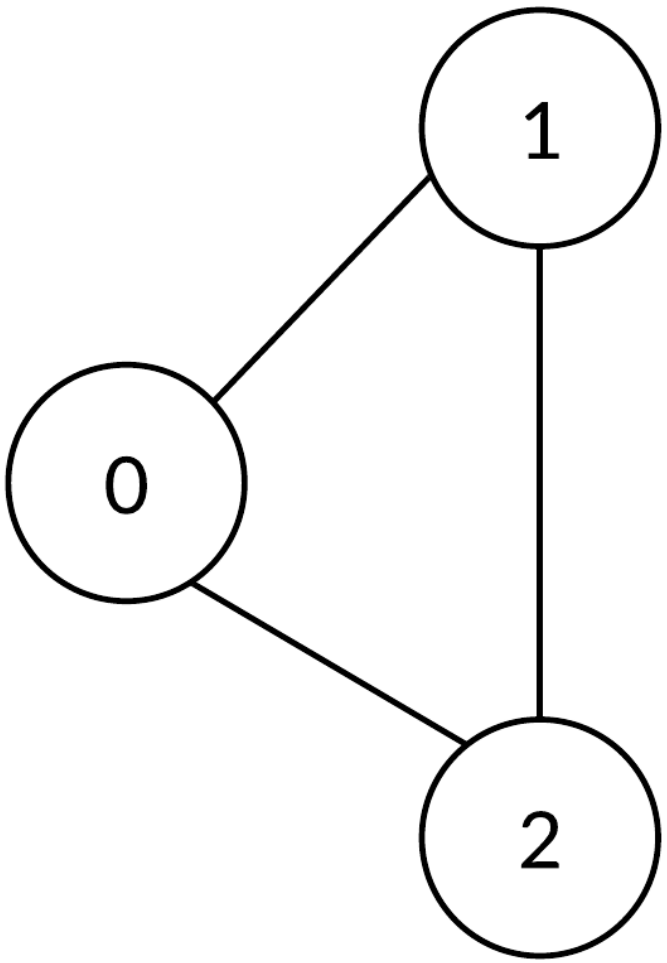
Um grafo G é definido como $G = (V, E)$

- $V = \{v_1, v_2, \ldots, v_n\}$ é o conjunto de vértices
- $E = \{e_1, e_2, \ldots, e_m\}$
  - $e_i = (u, v) \mid u, v \in V$

DIREÇÃO

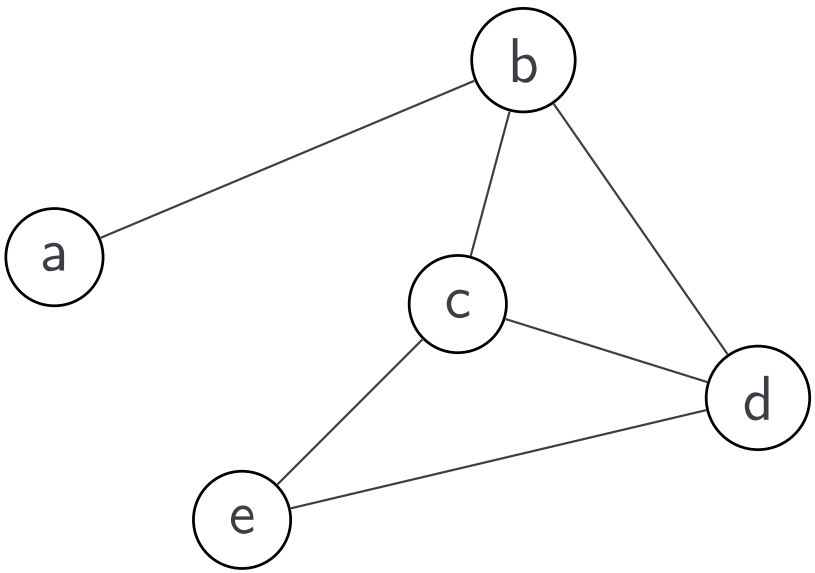
Um grafo pode ser direcionado ou não-direcionado

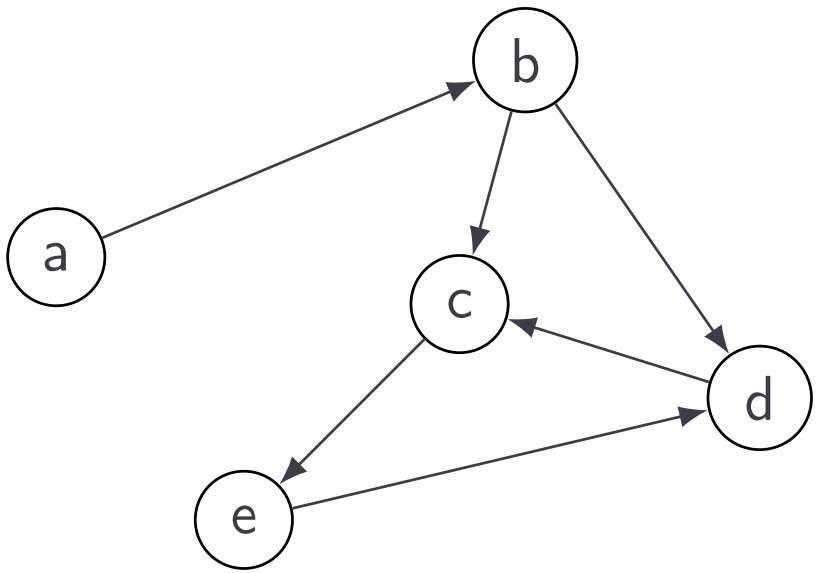




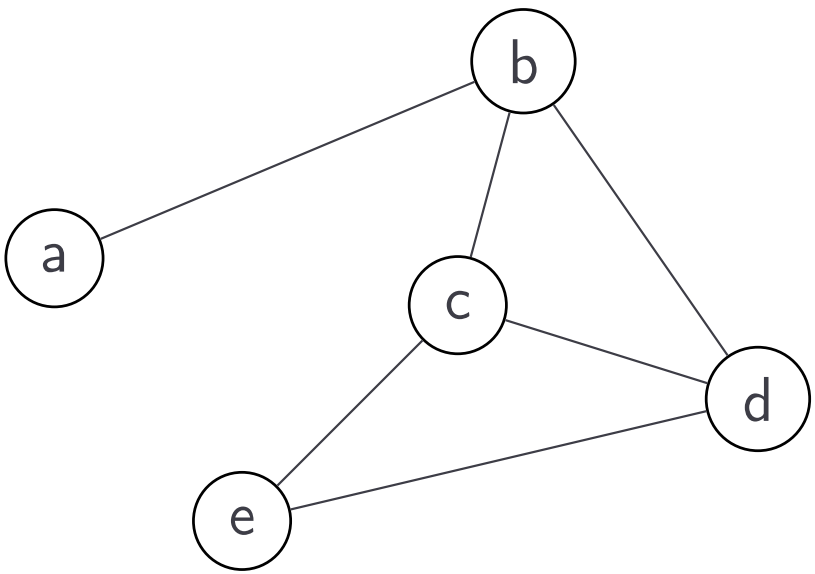
CAMINHOS E CICLOS

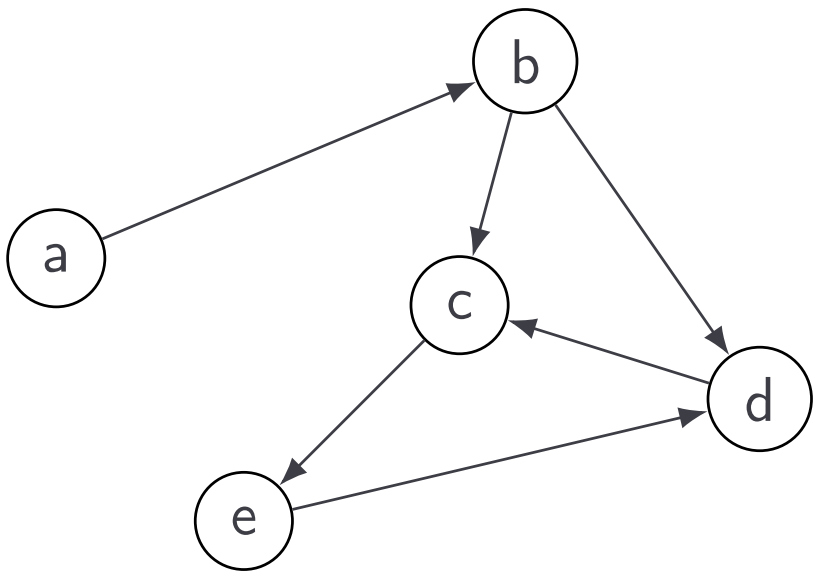
Caminho $C =< c, e, d, c >$





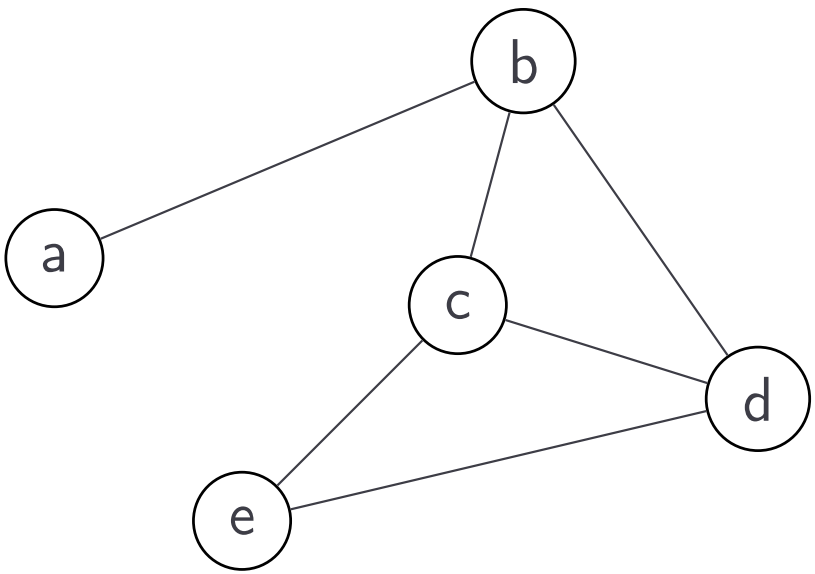
ADJACÊNCIA E GRAU

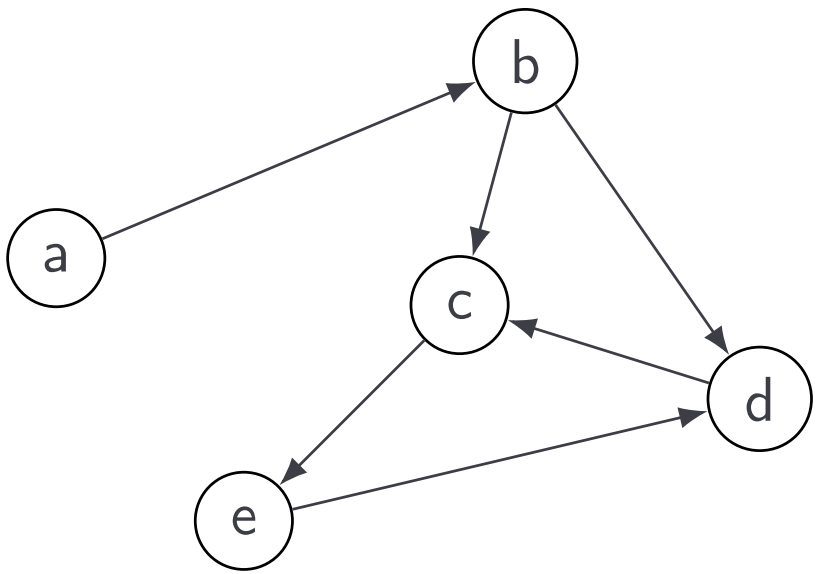




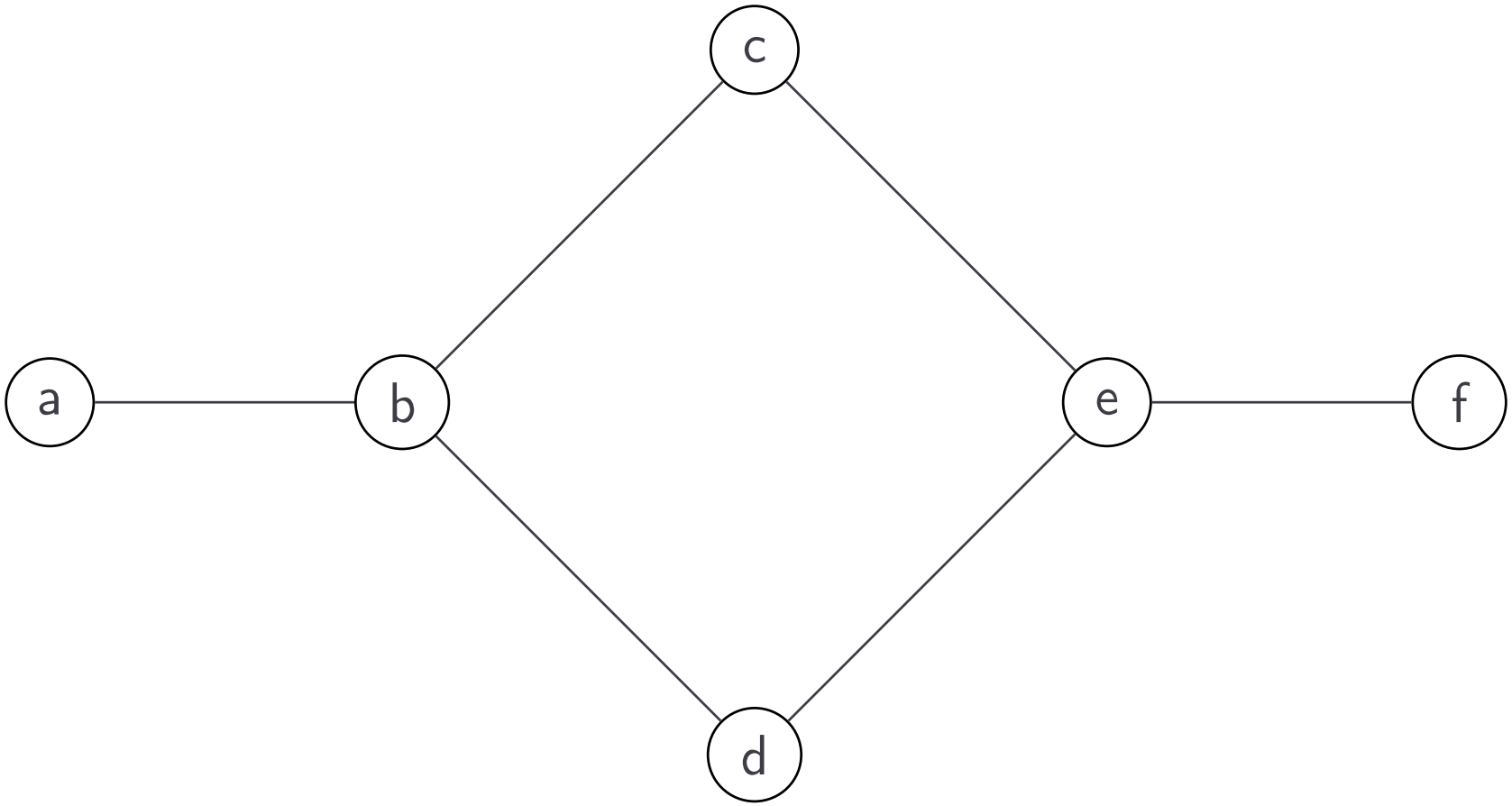
FECHO TRANSITIVO

Direto e inverso





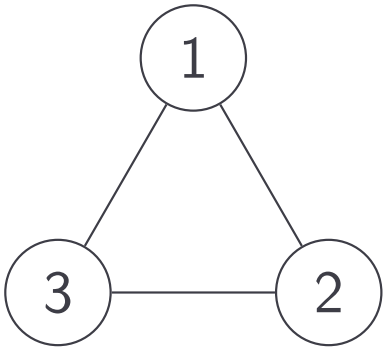
FONTE E SUMIDOURO



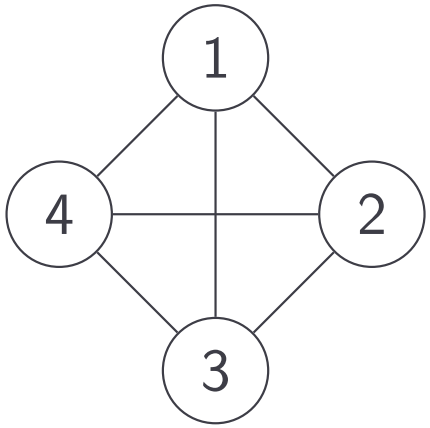
Fonte: a
Sumidouro: f

GRAFO COMPLETO

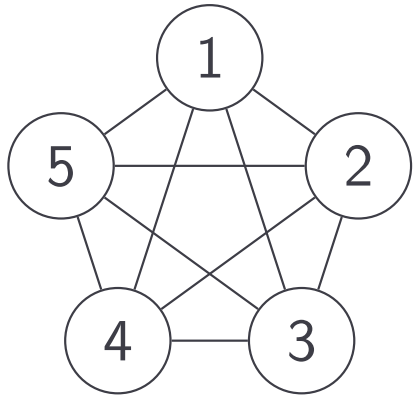
K_2



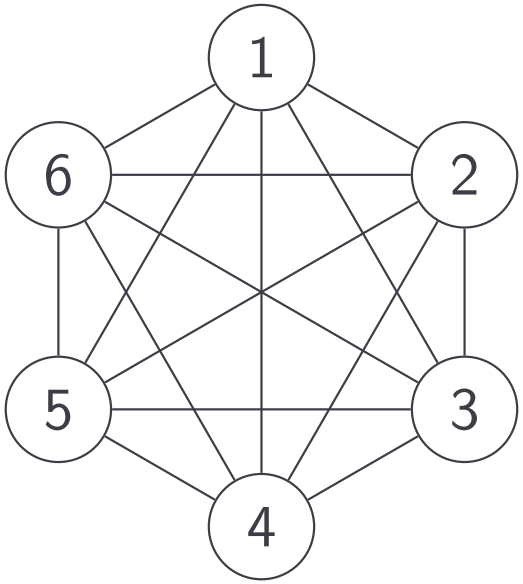
K_3



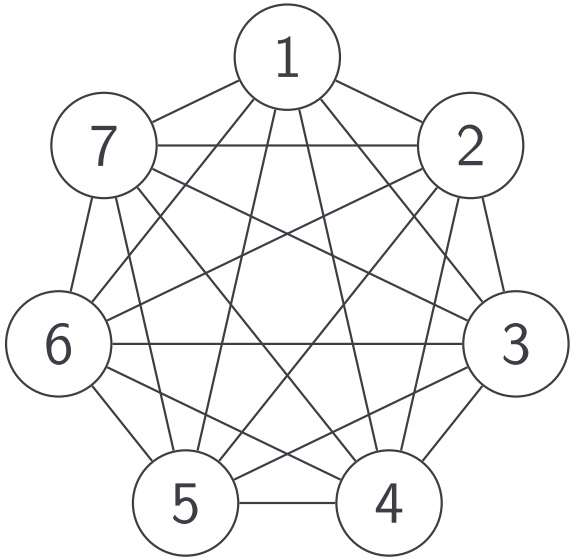
K_4



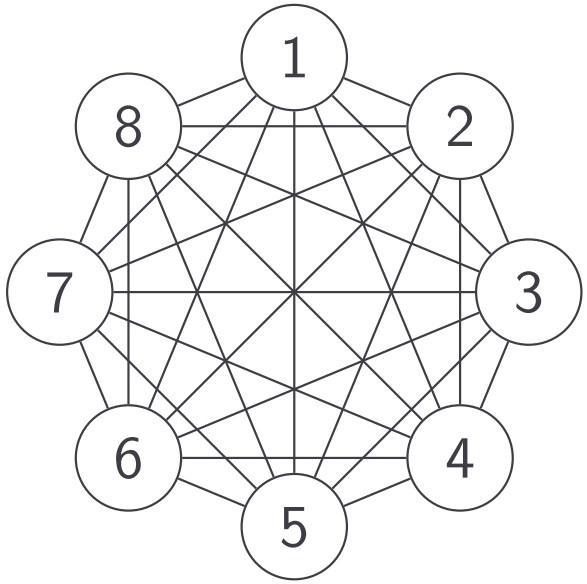
K_5



K_6

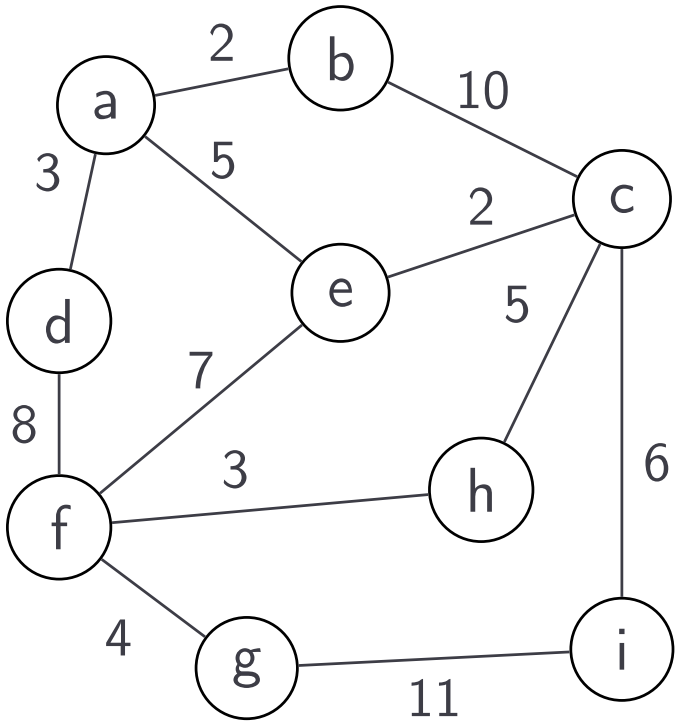


K_7

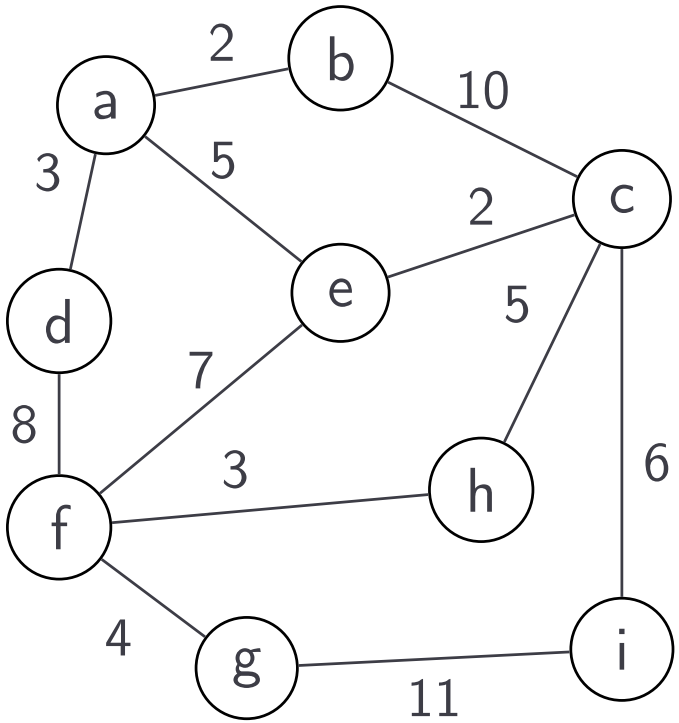


K_8

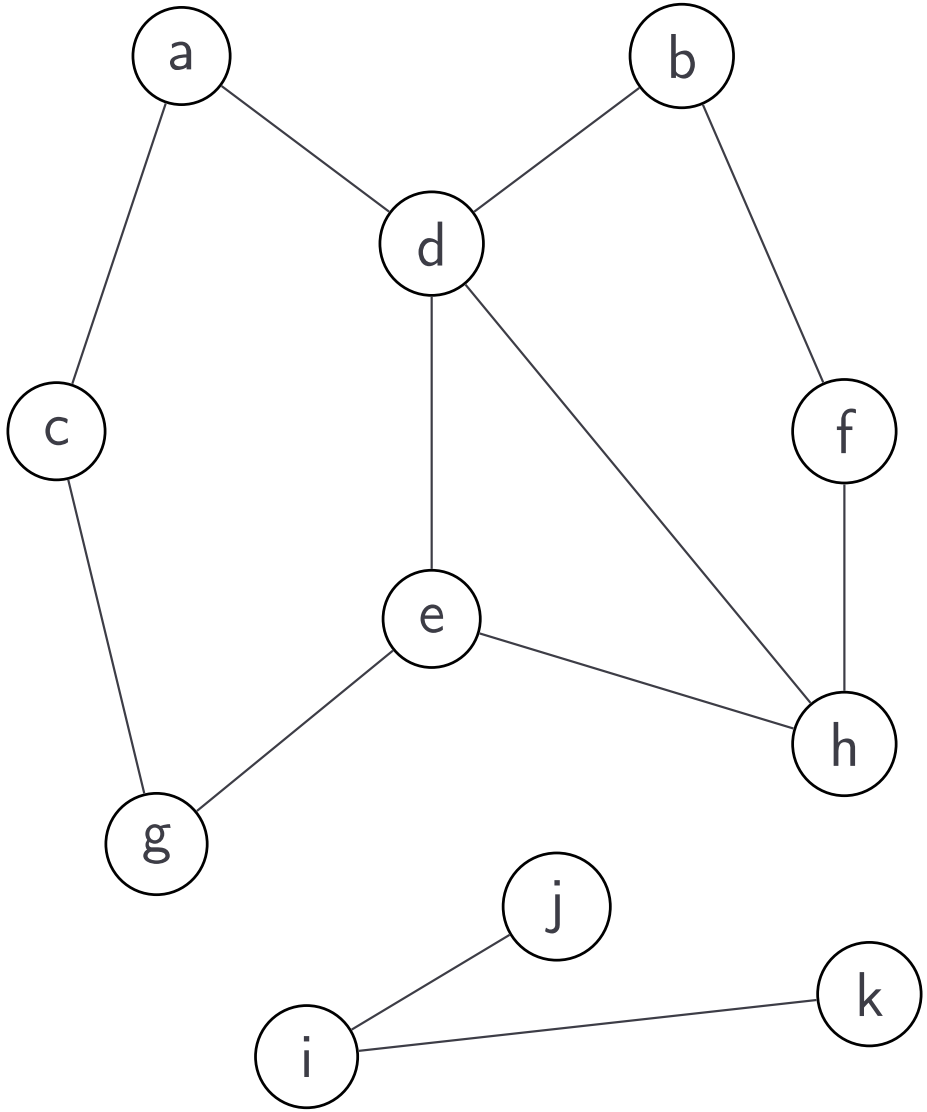
GRAFO COM PESOS



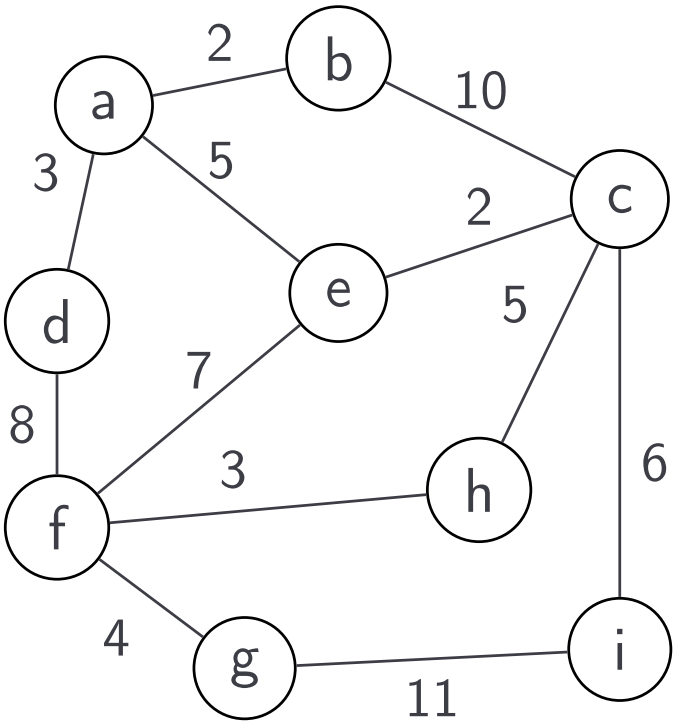
GRAFO CONEXO

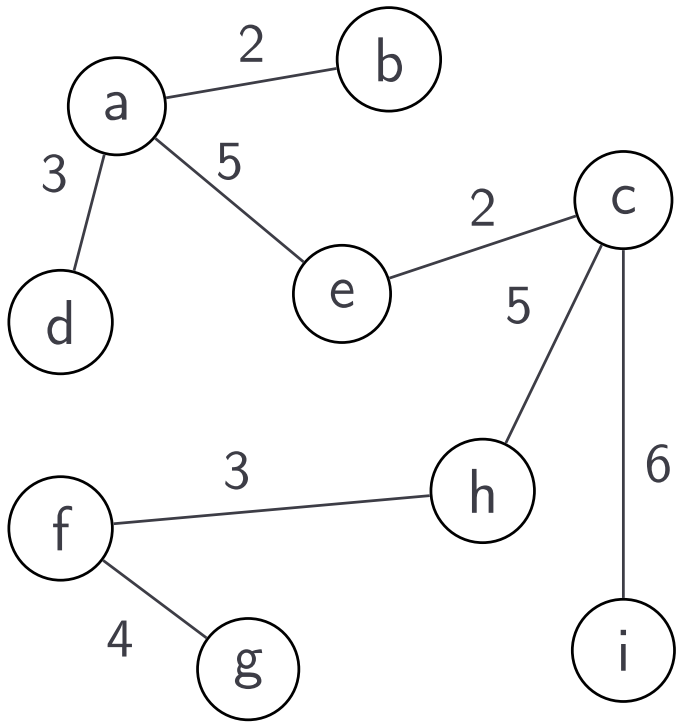


GRAFO DESCONECTADO E COMPONENTES CONEXAS

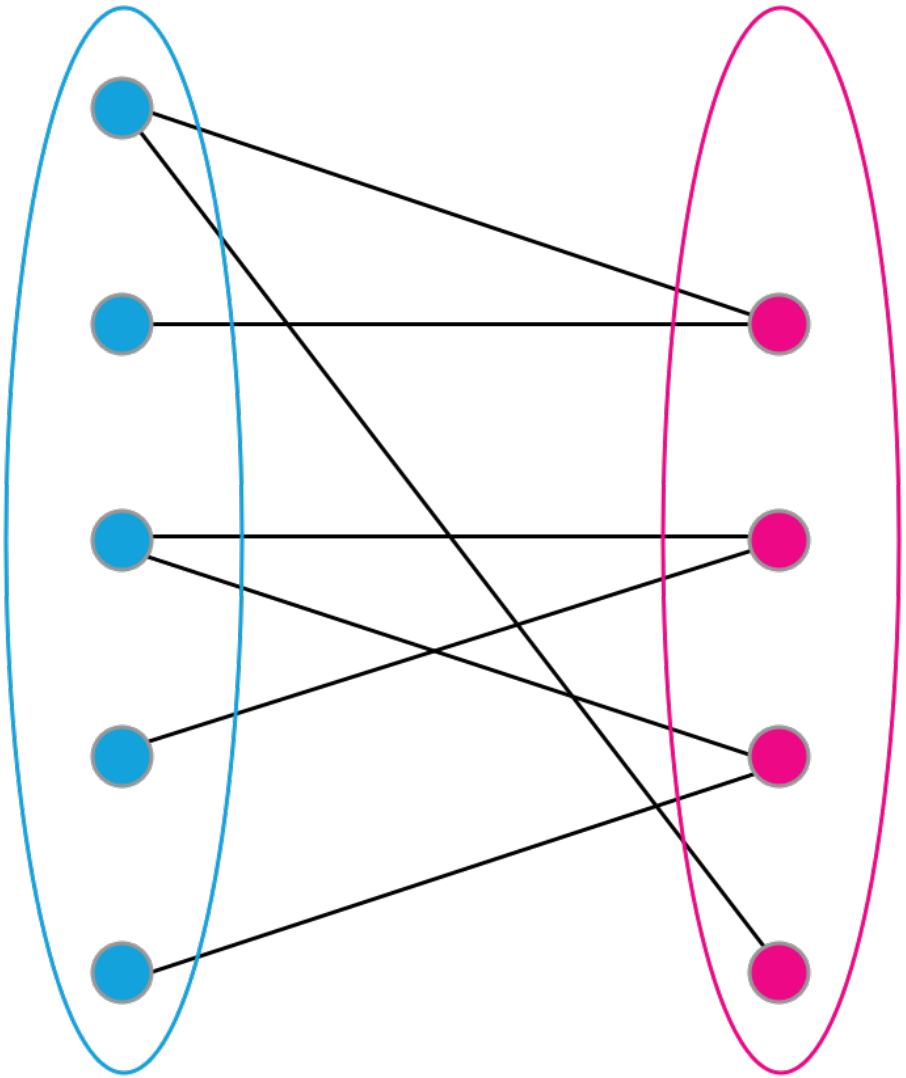


ÁRVORE GERADORA (MÍNIMA)





GRAFO BIPARTIDO



PROPRIEDADES ADICIONAIS

Diversas destas propriedades serão utilizadas no decorrer deste curso

Grafos são uma das estruturas mais importantes em Ciência da Computação, tendo aplicações em uma infinidade de áreas

- Redes
- Biologia
- Eletrônica
- Pesquisa Operacional
- ... ▸ Link

Interessados em um pouco mais de propriedades de grafos podem acessar o seguinte link ▸ Link

ESTRUTURAS DE DADOS

Existem duas estruturas de dados capazes de representar grafos

- Matriz de adjacência
- Lista de adjacência

Cada estrutura difere-se da outra pela complexidade de suas operações

- Complexidade de adicionar ou retirar nós
- Complexidade de inserir ou remover arestas
- Complexidade de pesquisa
  - Saber se uma aresta existe ou não
- Diferentes complexidades de espaço

MATRIZ DE ADJACÊNCIA

Talvez seja a maneira mais natural de se representar um grafo

- Grafo com n vértices
- Matriz bi-dimensional $n \times n$
- Complexidade de espaço: $\mathcal{O}(n^2) = \mathcal{O}(m)$

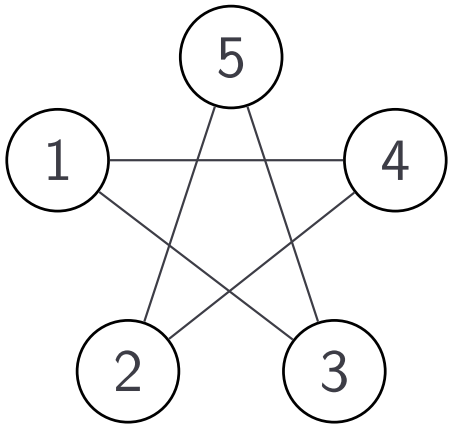
Inserção e remoção de vértices é cara

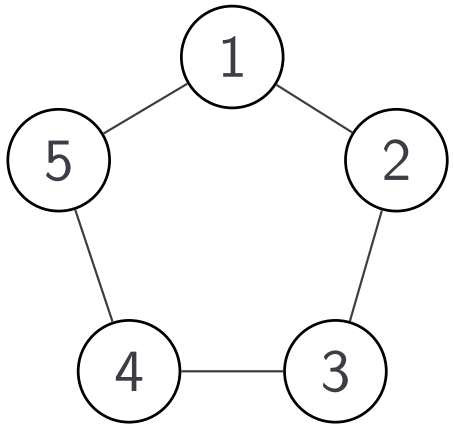
- Necessário alocar ou desalocar memória

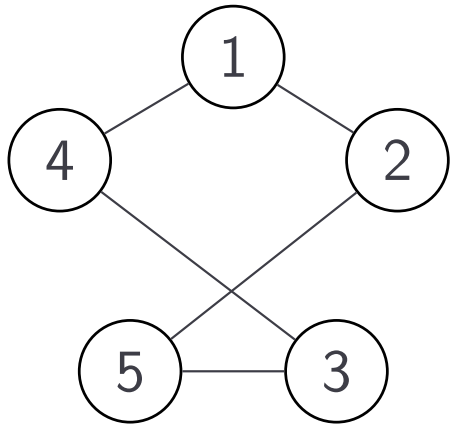
Modificação de arestas e pesquisa é barata

- Necessário apenas modificar (ou verificar) uma célula específica da matriz

MATRIZ DE ADJACÊNCIA







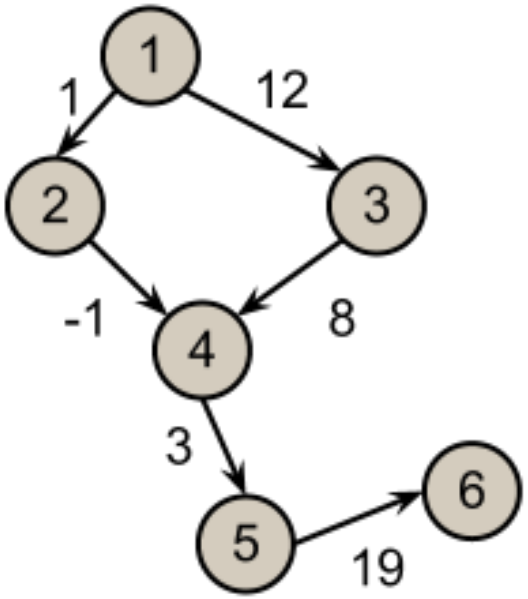
	1	**2**	**3**	**4**	**5**
1	0	0	1	1	0
2	0	0	0	1	1
3	1	0	0	0	1
4	1	1	0	0	0
5	0	1	1	0	0

	1	**2**	**3**	**4**	**5**
1	0	1	0	0	1
2	1	0	1	0	0
3	0	1	0	1	0
4	0	0	1	0	1
5	1	0	0	1	0

	1	**2**	**3**	**4**	**5**
1	0	1	0	1	0
2	1	0	0	0	1
3	0	0	0	1	1
4	1	0	1	0	0
5	0	1	1	0	0

MATRIZ DE ADJACÊNCIA

Weighted Directed Graph & Adjacency Matrix



Weighted Directed Graph

	1	2	3	4	5	6
1	0	1	12	0	0	0
2	-1	0	0	-1	0	0
3	-12	0	0	8	0	0
4	0	1	-8	0	3	0
5	0	0	0	-3	0	19
6	0	0	0	0	-19	0

Adjacency Matrix

LISTA DE ADJACÊNCIA

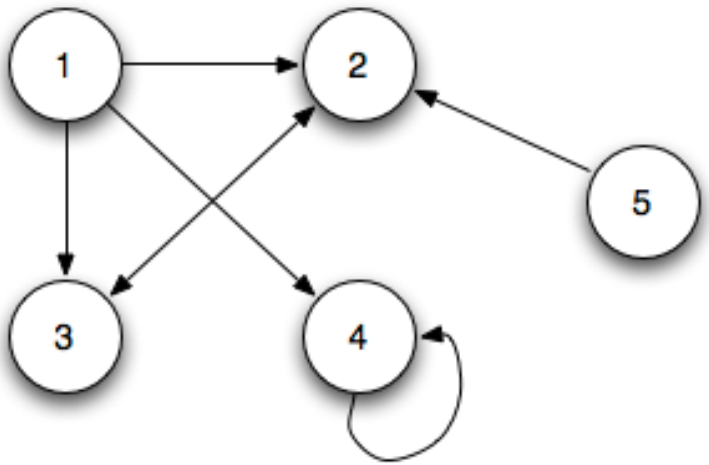
Uma lista de adjacência pode ser representada como uma lista de listas

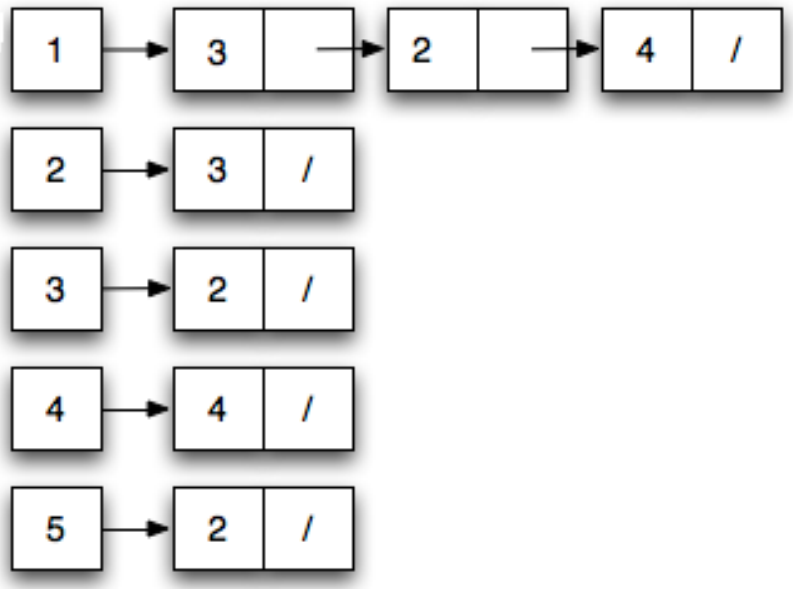
- Uma lista que contém todos os vértices do grafo
- Cada lista contém outra lista
  - Contém todos os vértices adjacentes

Complexidades diferem das de matriz de adjacência

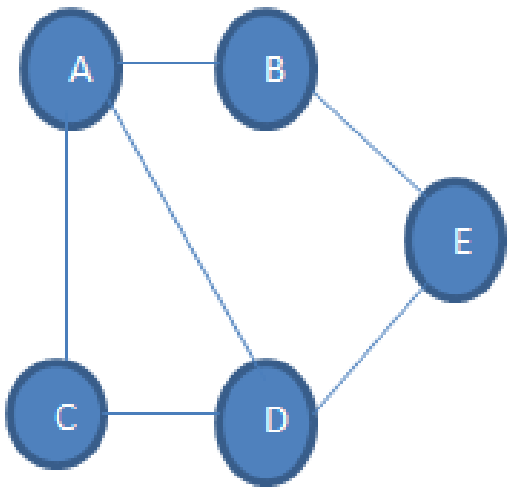
- Complexidade de espaço: $\mathcal{O}(n^2) = \mathcal{O}(m)$
- Inserção, pesquisa e remoção de arestas: $\mathcal{O}(n)$
- Inserção e remoção de vértices: $\mathcal{O}(1)$

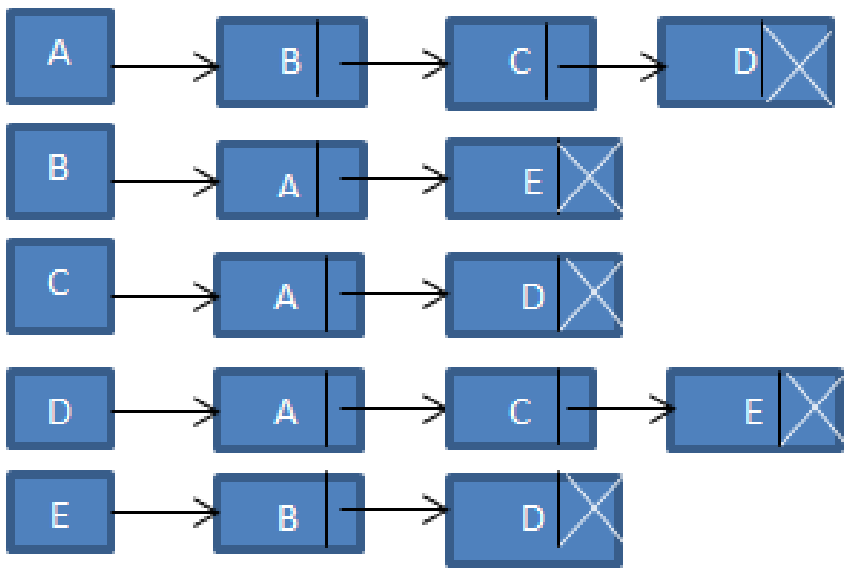
LISTA DE ADJACÊNCIA



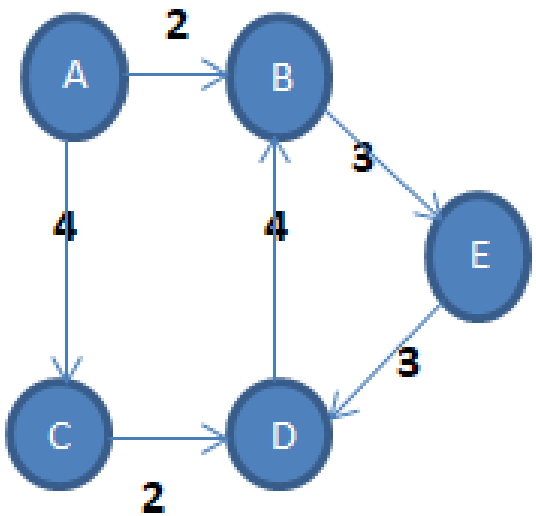


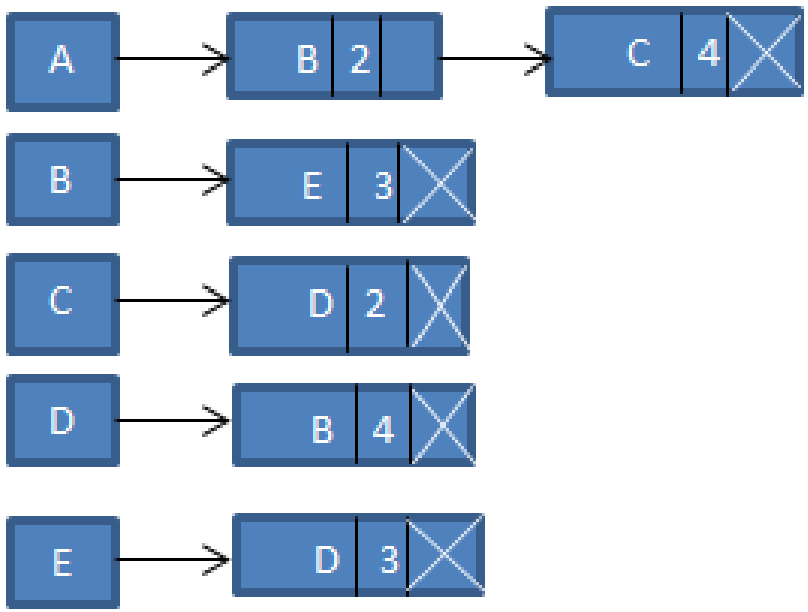
LISTA DE ADJACÊNCIA





LISTA DE ADJACÊNCIA





INTRODUÇÃO A OTIMIZAÇÃO

OTIMIZAÇÃO LINEAR

Esse curso foca em heurísticas e metaheurísticas para otimização

- Desta forma, é interessante também termos uma introdução a otimização

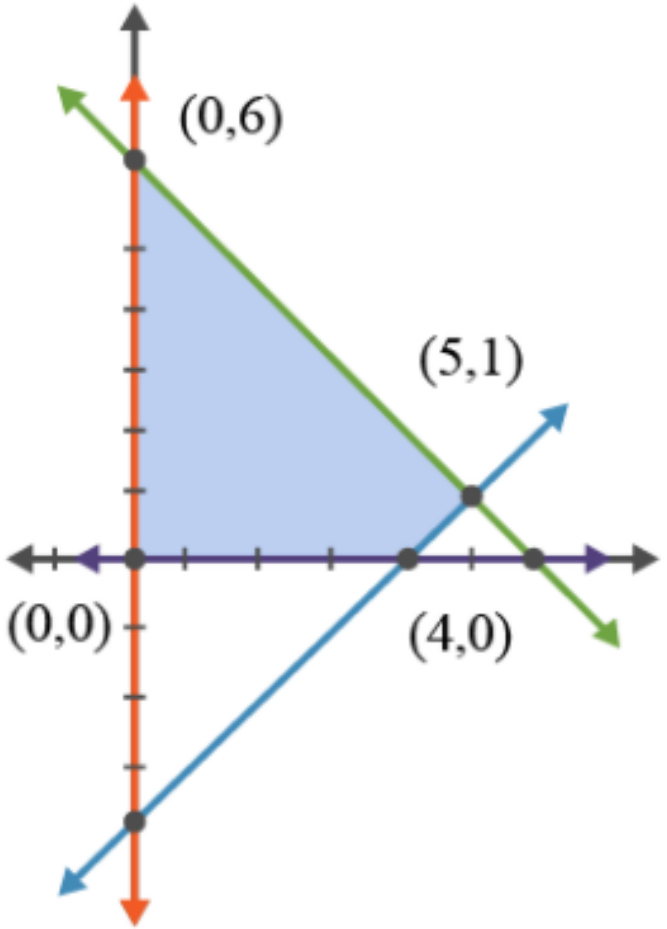
O objetivo da otimização linear é resolver sistemas lineares

- Uma (ou mais) funções objetivo
- Conjunto de restrições
- Variáveis no domínio dos reais ($\mathbb{R}$)

Deve-se atribuir um valor para cada uma das variáveis do problema de tal forma que

- A função objetivo seja minimizada (ou maximizada)
- Todas as restrições sejam respeitadas

$$
\begin{array}{lrl}
\min & 2x + y & \\
& x + y & \leq 6 \\
& x - y & \leq 4 \\
& x & \geq 0 \\
& y & \geq 0
\end{array}
$$



OTIMIZAÇÃO LINEAR

O principal uso de modelos de programação linear é para otimizar (encontrar o mínimo ou o máximo) de algo

- Maximizar o lucro
- Minimizar as perdas
- Minimizar o tempo gasto
- Minimizar número de funcionários
- Maximizar o número de produtos produzidos
- Minimizar gasto de combustível
- ...

OTIMIZAÇÃO LINEAR

Modelos de otimização linear normalmente tentam representar um problema de mundo real através de um sistema de equações

A **função objetivo** representa aquilo que você quer otimizar

- Minimizar ou maximizar

As **variáveis** representam a tomada de decisão

- Vou utilizar esta rota ou aquela?
- Quantos produtos deste tipo eu vou produzir?

As **restrições** representam as limitações existentes

- Qual é o número máximo de horas por dia que estes funcionários podem trabalhar?
- Quantos metros cúbicos de madeira eu tenho para produzir estes móveis?
- Quantos caminhões eu possuo para fazer entregas?

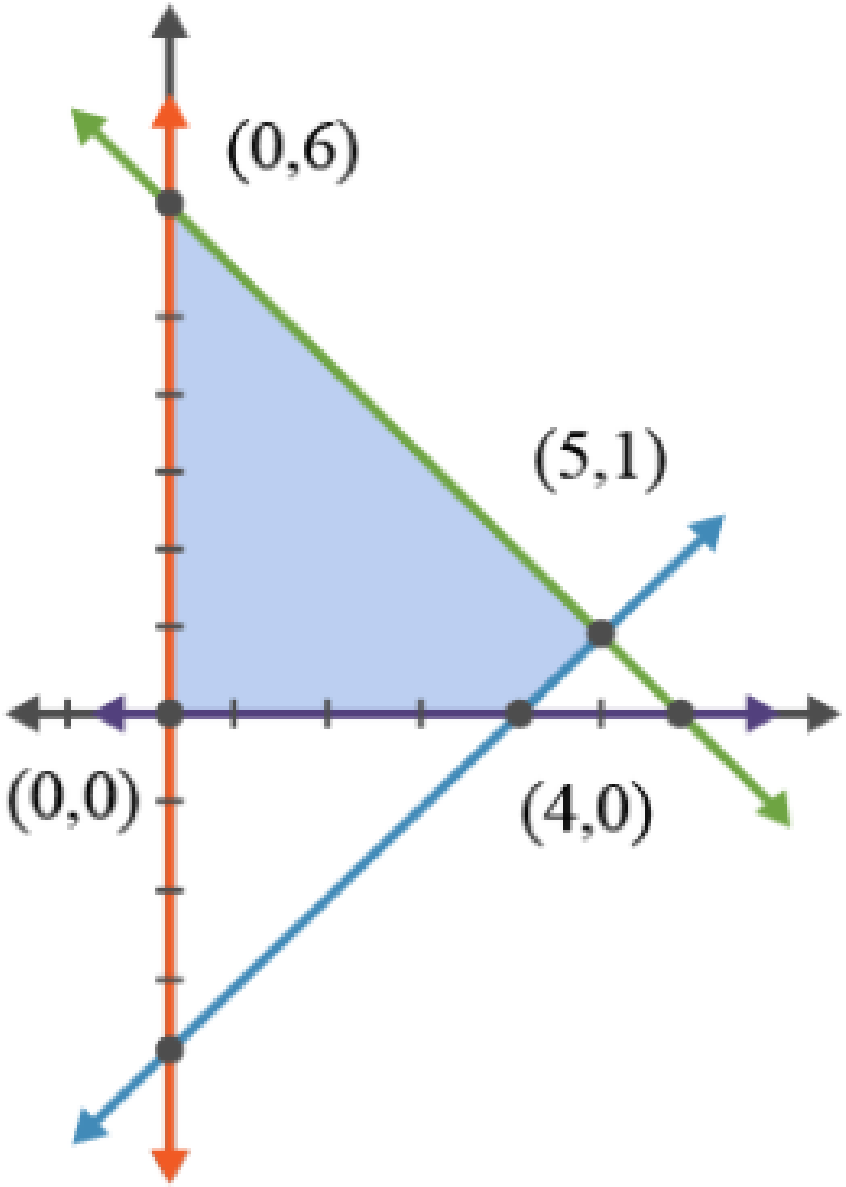
$$
\begin{aligned}
\min \quad & 2x + y & \\
& x + y & \leq 6 \\
& x - y & \leq 4 \\
& x & \geq 0 \\
& y & \geq 0
\end{aligned}
$$

Função objetivo

Restrições

$$
\begin{aligned}
\min \quad & 2\mathbf{x} + \mathbf{y} \\
& \mathbf{x} + \mathbf{y} && \leq 6 \\
& \mathbf{x} - \mathbf{y} && \leq 4 \\
& \mathbf{x} && \geq 0 \\
& \mathbf{y} && \geq 0
\end{aligned}
$$

Direção da função objetivo
Função objetivo
Restrições
Variáveis (em negrito)
Restrições de domínio das variáveis



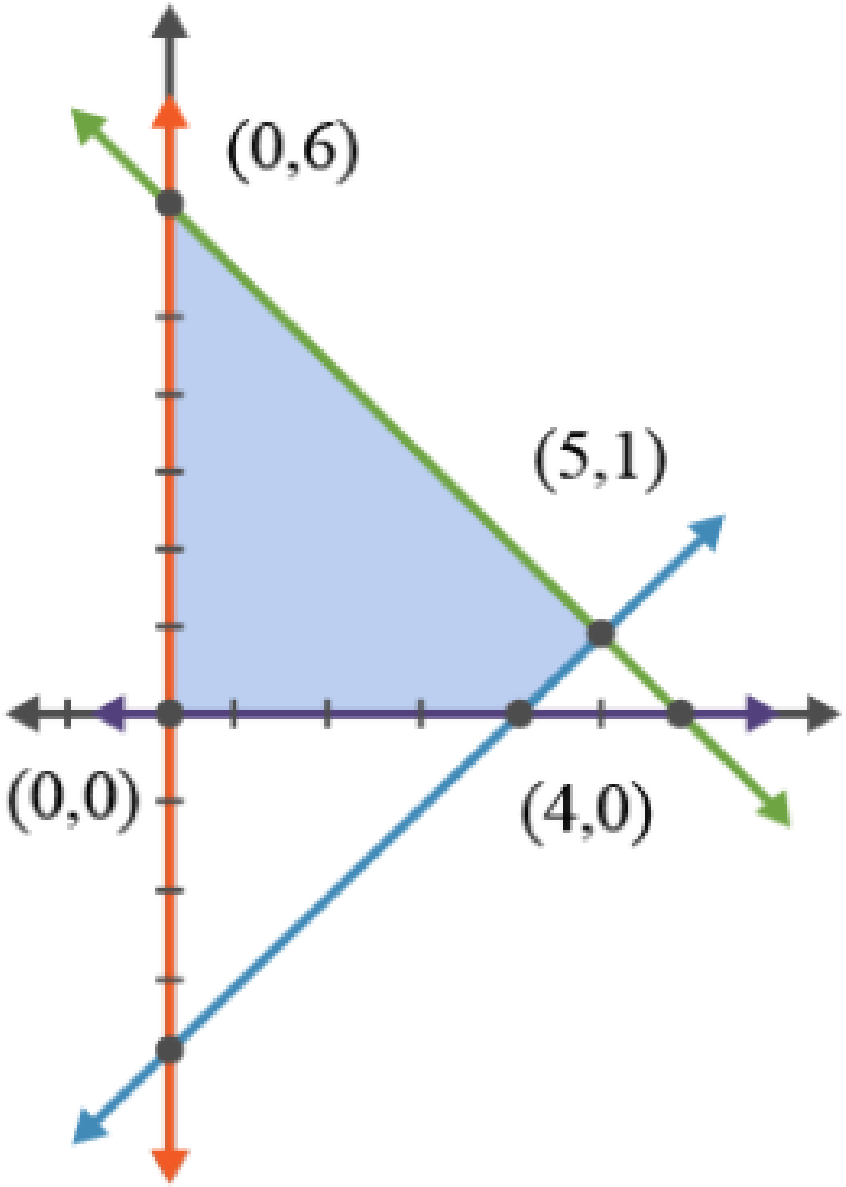
$x + y \leq 6$

$x - y \leq 4$

$y > 0$

$x > 0$

Soluções viáveis

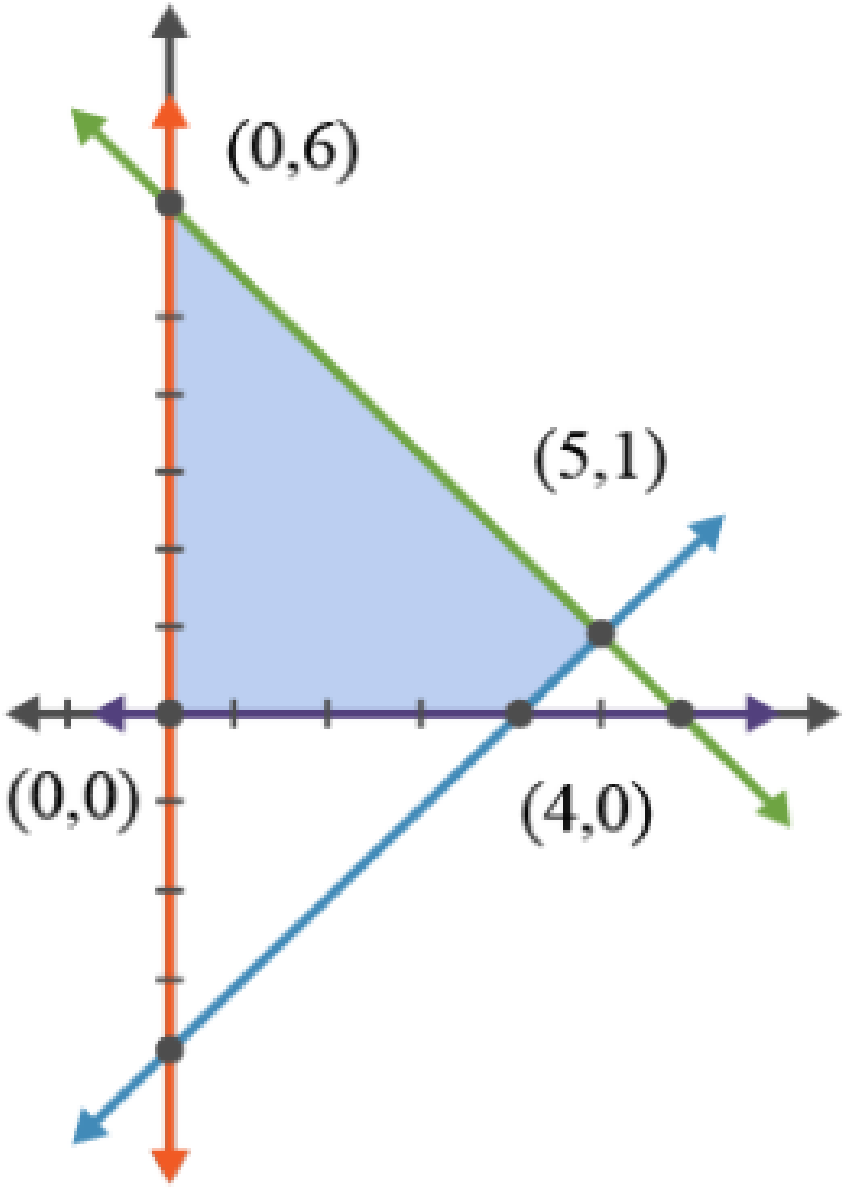


O espaço azul representa o conjunto de soluções viáveis de nosso problema

- Soluções ótimas
- Soluções sub-ótimas

Solução ótima está em um vértice

- Encontro de duas ou mais restrições



$$\min \quad 2x + y$$

x	y	resultado
0	0	0
0	6	6
5	1	11
4	0	8

EXEMPLO DE MODELAGEM

Uma fábrica produz dois produtos, A e B.

- Cada um deve ser processado por duas máquinas M_1 e M_2

Devido à programação de outros produtos que também usam estas máquinas, estão disponíveis para os produtos A e B apenas 24 horas da máquina M_1 e 16 horas da máquina M_2.

EXEMPLO DE MODELAGEM

Para produzir uma unidade do produto A são necessárias

- 4 horas da máquina M_1
- 4 horas da máquina M_2

Para produzir uma unidade do produto B são necessárias

- 6 horas da máquina M_1
- 2 horas da máquina M_2

O produto A é vendido com um lucro de R$ 80,00, enquanto o produto B é vendido com um lucro de R$ 60,00

EXEMPLO DE MODELAGEM

Existe uma previsão de demanda máxima de 3 unidades para B, mas nenhuma restrição de demanda para A.

Deseja-se saber: quanto produzir de cada produto para maximizar o lucro?

Produto	**Horas de M_1**	**Horas deM_2**	**Demanda Max**	**Lucro Unitário**
A	4	4	-	80
B	6	2	3	60
Horas Disp.	24	16	-	-

EXEMPLO DE MODELAGEM

$$
\begin{aligned}
\max \quad & 80\mathbf{Xa} + 60\mathbf{Xb} \\
& 4\mathbf{Xa} + 6\mathbf{Xb} \leq 24 \\
& 4\mathbf{Xa} + 2\mathbf{Xb} \leq 16 \\
& \mathbf{Xb} \leq 3 \\
& \mathbf{Xa} \geq 0 \\
& \mathbf{Xb} \geq 0
\end{aligned}
$$

Direção da função objetivo

Função objetivo

Restrições

Variáveis (em negrito)

Restrições de domínio das variáveis

PROGRAMAÇÃO INTEIRA

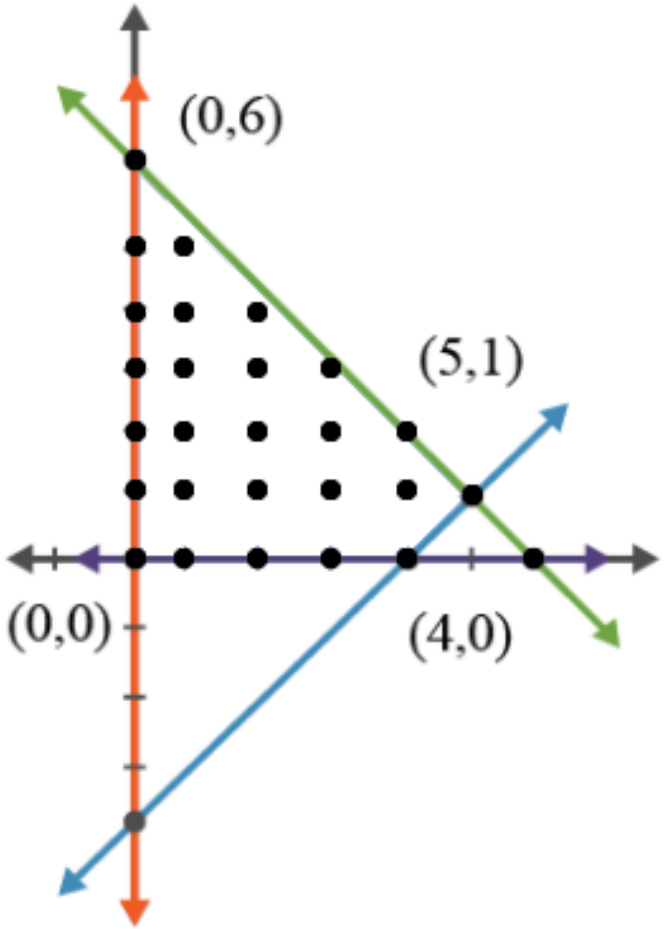
O objetivo da otimização linear é resolver sistemas lineares

- Uma (ou mais) funções objetivo
- Conjunto de restrições
- Variáveis no domínio dos inteiros ($\mathbb{Z}$)

Deve-se atribuir um valor para cada uma das variáveis do problema de tal forma que

- A função objetivo seja minimizada (ou maximizada)
- Todas as restrições sejam respeitadas

$$
\begin{array}{lll}
\min & 2x + y & \\
& x + y & \leq 6 \\
& x - y & \leq 4 \\
& x & \in \mathbb{Z} \\
& y & \in \mathbb{Z}
\end{array}
$$



MODELAGEM DE PROBLEMAS COMO PROGRAMAÇÃO INTEIRA

É possível modelar problemas não-polinomiais

- Algoritmos de programação inteira são não-polinomiais
- Exploram um número exponencial de soluções
  - Baseado nas possíveis combinações de valores das variáveis inteiras

Exemplos de problemas

- Problema do caminho máximo
- Problema da cobertura de conjuntos
- Problema da mochila binária
- Problema da árvore de Steiner
- Problema da árvore geradora minima restrita em grau
- Problema da satisfabilidade

Nesta disciplina vamos trabalhar, marjoritariamente, com problemas de programação inteira