



PESQUISA OPERACIONAL II

PROBLEMAS DE MINIMIZAÇÃO DE REDES: ALGORITMO DE PRIM

Prof. Dr. Daniel Caetano

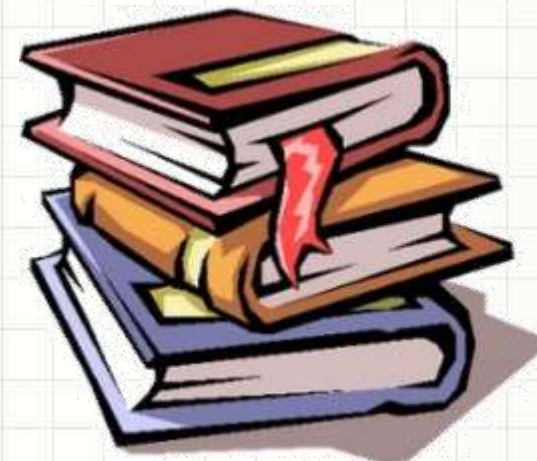
2019 - 1

Objetivos

- Tomar contato com o conceito de árvores
- Conhecer o que é uma árvore geradora
- Compreender a utilidade de árvores geradoras mínimas
- Aprender a determinar árvores geradoras mínimas
- **Atividade Aula 2 – SAVA!**



Material de Estudo



Material

Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>
(Pesquisa Operacional II – Aula 2)

Minha Biblioteca

Introdução à Pesquisa Operacional
(Hillier/Lieberman), Cap. 9, Seção 9.4

Biblioteca Virtual

Grafos e Redes: Teoria e Algoritmos Básicos, Cap. 6,
seção 6.6

Recursos na Web

https://www.ime.usp.br/~pf/algoritmos_para_grafos/index.html#mst

Antes de Mais nada...

- **Não deixe de consultar o material da 1ª Aula!**
- **Otimize seus estudos**
 - Toda semana acessar o SAVVA!
 - Se preparar para conteúdo da semana seguinte!
- **Exercícios Semanais**
 - Exercícios propostos a cada aula: SAVVA
- **Será controlada a presença**
 - Chamada ocorrerá sempre às 20:30 / 22:25
 - Nome fora da lista = falta

- **Contato**

Professor

Informações de Contato

Daniel Caetano

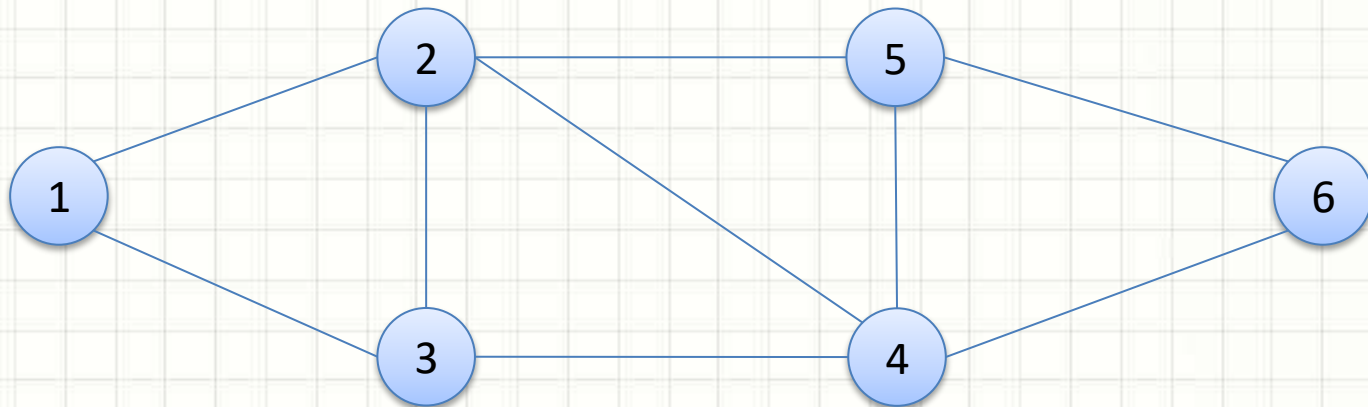
prof@caetano.eng.br



RETOMANDO:
GRAFOS E REDES

Grafos – Uma Noção Simplificada

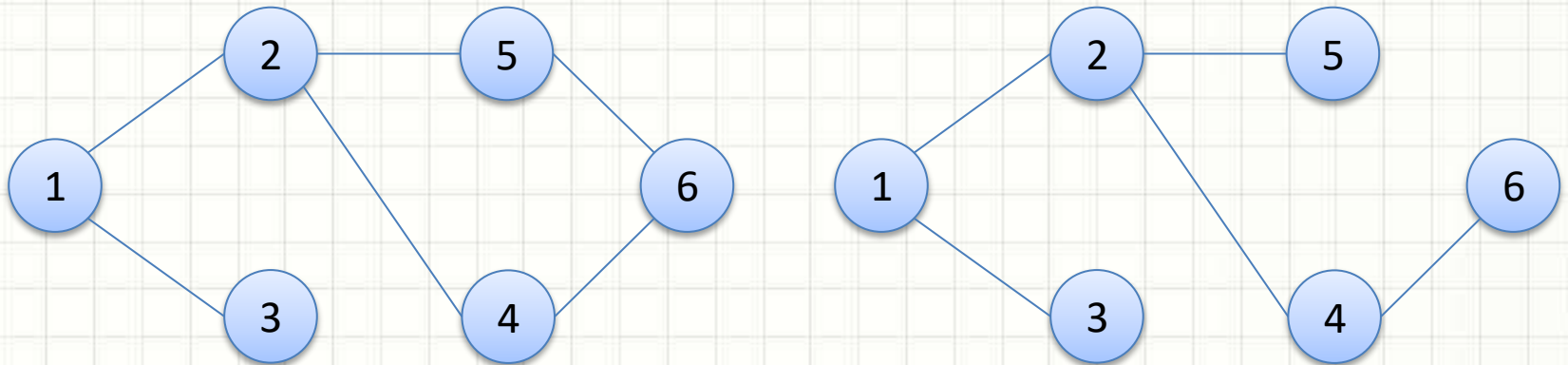
- São *estruturas abstratas* compostas por:
 - Um conjunto de nós **N**
 - Um conjunto de arcos **A**, que ligam esses nós



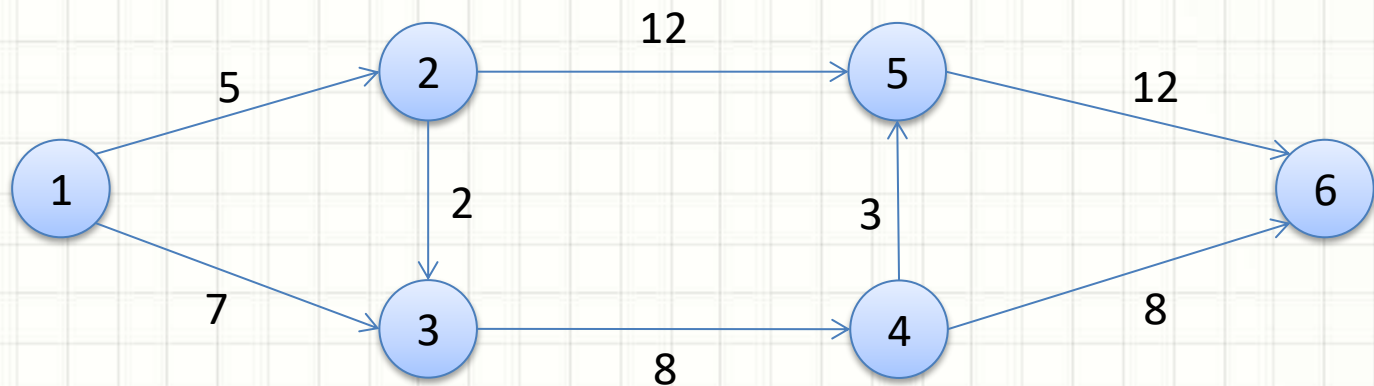
- Os grafos podem ser direcionados ou não

Grafos Cíclicos

- Grafos podem formar ou não ciclos



- Grafos valorados: redes

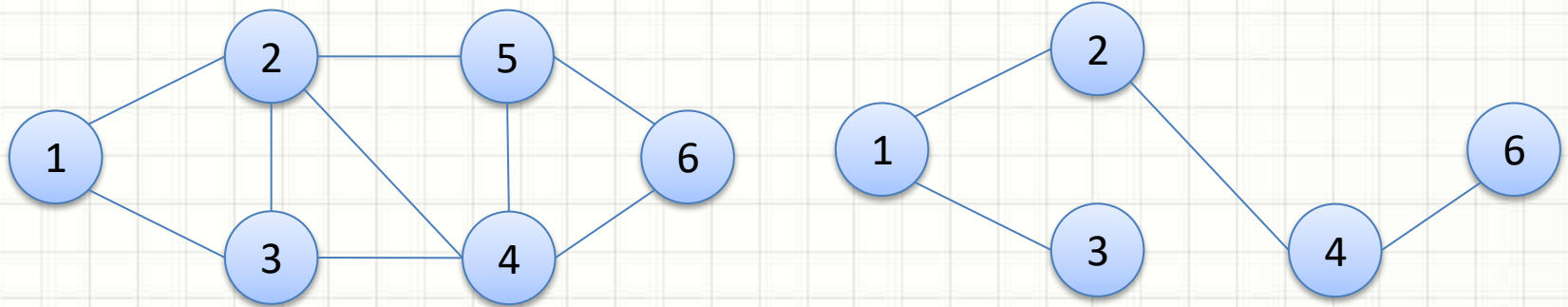




FUNDAMENTOS:
**ÁRVORES GERADORAS
DE CUSTO MÍNIMO**

O que é uma Árvore?

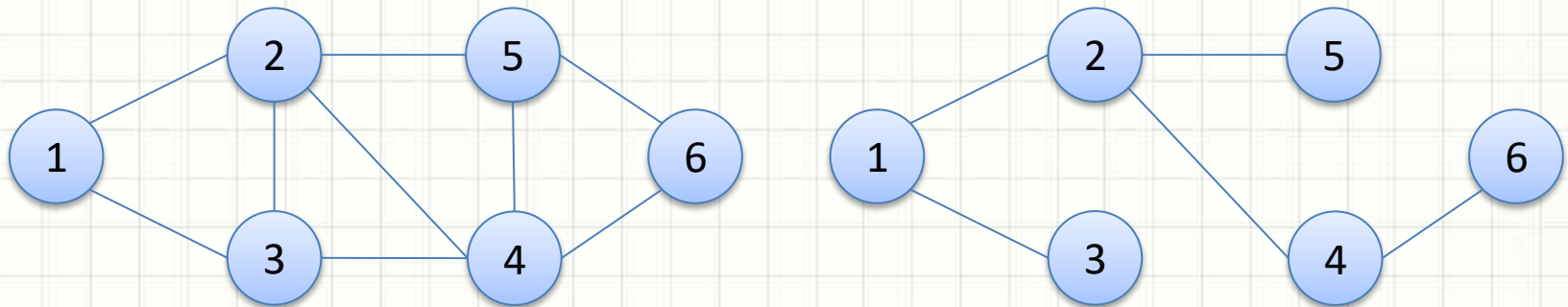
- É um sub-grafo onde não há ciclos



- Não é necessário
 - Preservar todos os nós
 - Preservar todos os arcos
- Só há um caminho entre dois nós da árvore

O que é uma Árvore Geradora?

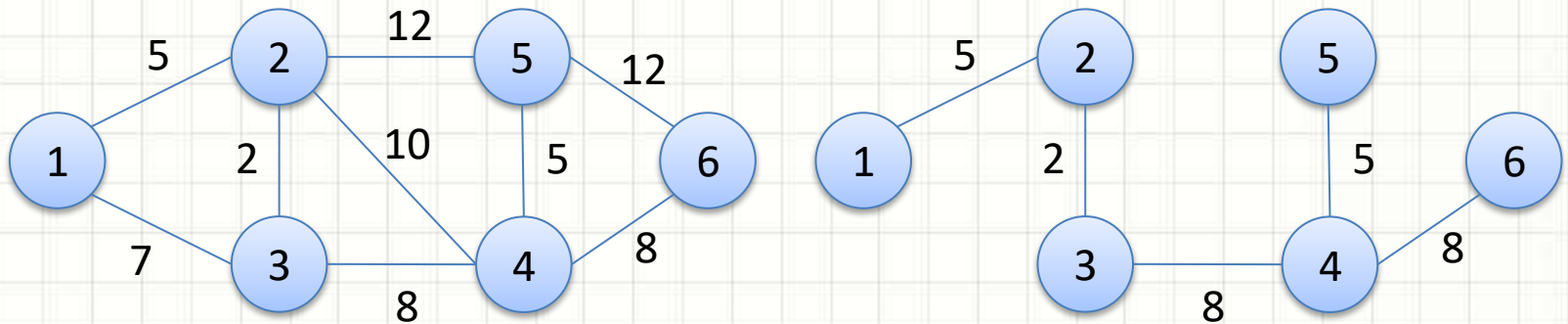
- É uma árvore de um grafo em que:
 - Não são preservados todos os arcos...
 - Mas preserva todos os nós



- Se o grafo original tem N nós...
 - A árvore geradora sempre terá $N-1$ arcos

Árvore Geradora de Custo Mínimo

- Se os arcos são valorados
 - Existe uma A.G. cuja soma dos valores é mínima
 - Essa é a Árvore Geradora de Custo Mínimo



- Em inglês “Minimum Spanning Tree”
- Existem algoritmos para encontrá-la
- Mas... para que serve isso?

Árvore Geradora de Custo Mínimo

- Usos comuns na engenharia
 - Redes de telefonia fixa e TV a cabo
 - Redes de distribuição de energia elétrica
 - Redes de distribuição de água
 - Identificação de caminho com gargalo máximo
- Outros usos
 - Análise de agrupamento (Cluster Analysis)
 - Identificação de rostos em tempo real
 - Quando for necessário evitar ciclos em uma rede
 - ...

Determinando AGCM/MST

- Existem várias formas
- Modelagem Matemática

$$\min \sum_{ij \in E} c_{ij} x_{ij}$$

Sujeito a: $\sum_{ij \in E} x_{ij} = n - 1$

$$\sum_{ij \in E: i \in S, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V$$

$$x_{ij} \in \{0, 1\} \quad \forall ij \in E$$

- Algoritmos Práticos
 - Algoritmo de Prim
 - Algoritmo de Kruskal



O ALGORITMO DE PRIM

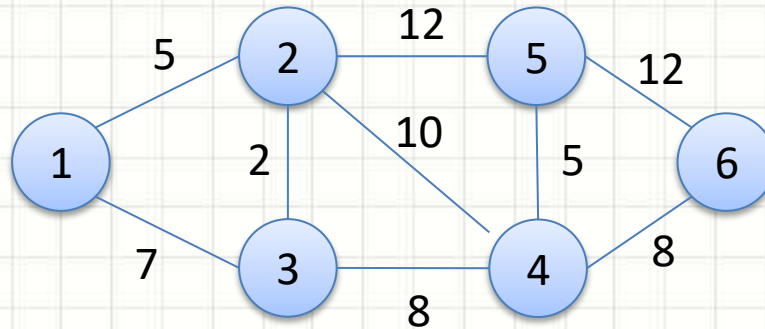
Algoritmo de Prim

- O algoritmo de Prim é bastante simples
 1. No grafo original **O**, escolha um nó **A** ligado ao **arco** de menor custo
 2. Transfira **A** para a árvore destino **D**
 3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**, desde que **B** não esteja em **D**
 4. Transfira **B** e **L** para **D**
 5. Se ainda há **nós** em **O** que não estão em **D**, volte ao passo 3.

Aplicando o Algoritmo de Prim

1. No grafo original **O**, escolha um nó **A** ligado ao **arco** de menor custo

O

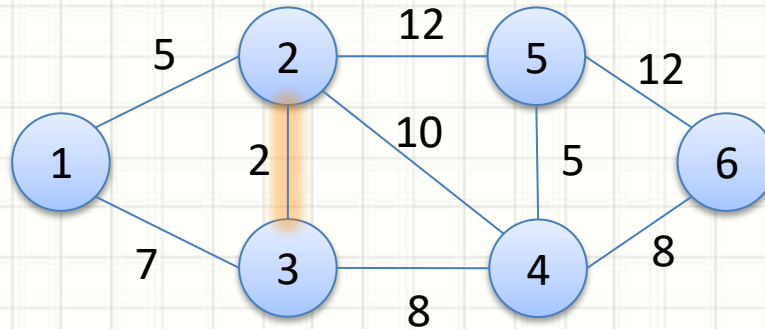


D

Aplicando o Algoritmo de Prim

1. No grafo original **O**, escolha um nó **A** ligado ao **arco** de menor custo

O

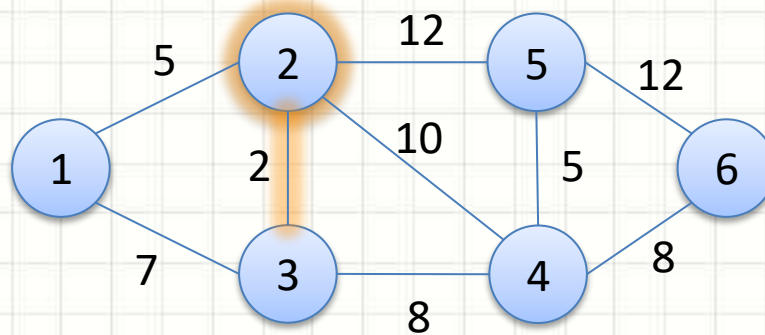


D

Aplicando o Algoritmo de Prim

1. No grafo original **O**, escolha um nó **A** ligado ao **arco** de menor custo

O

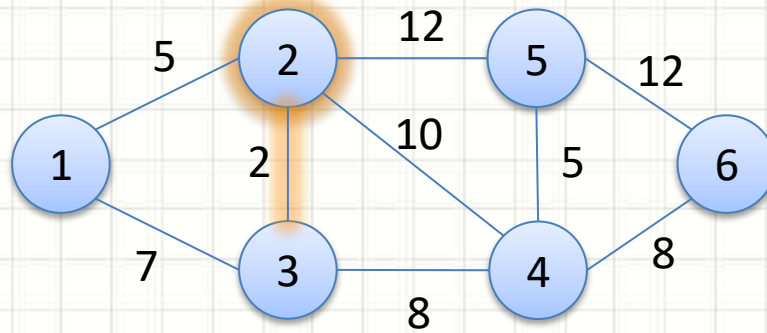


D

Aplicando o Algoritmo de Prim

2. Transfira **A** para a árvore destino **D**

O

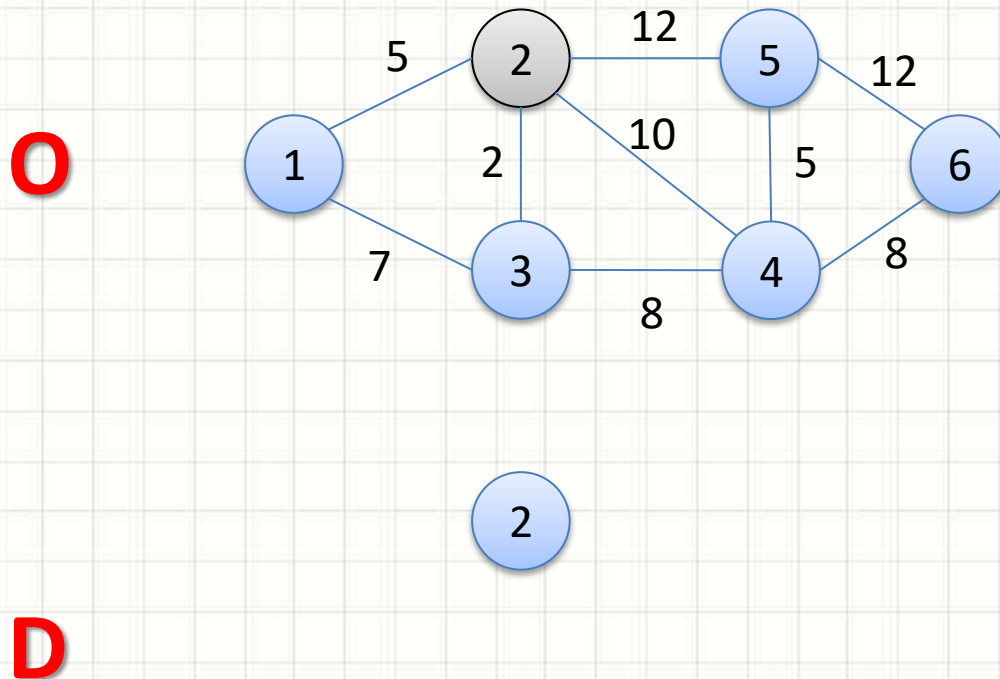


D



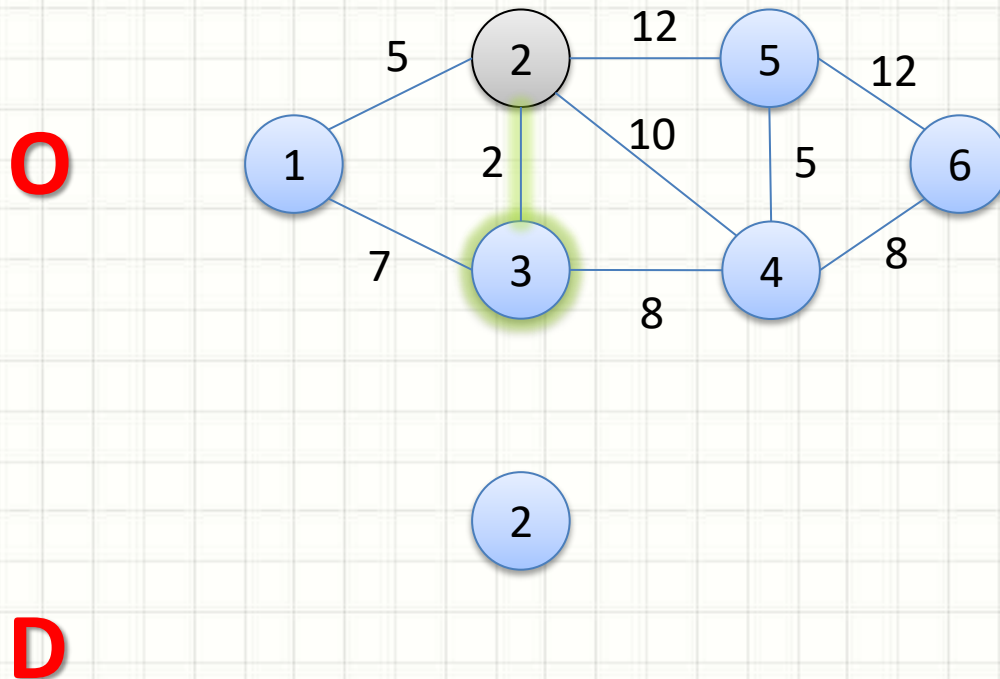
Aplicando o Algoritmo de Prim

3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

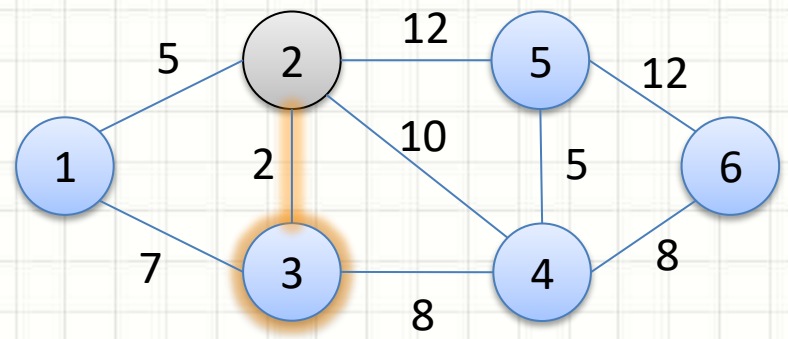
3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



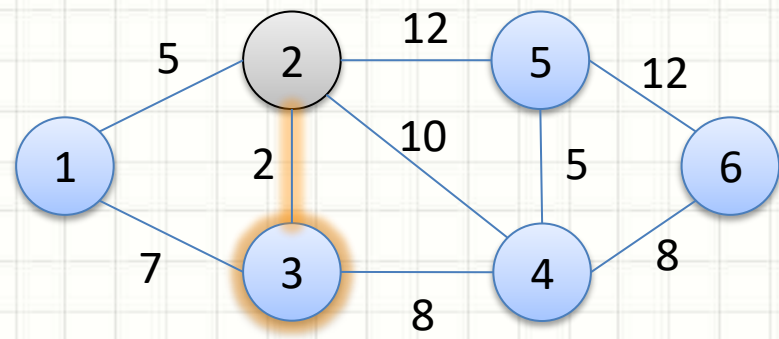
D



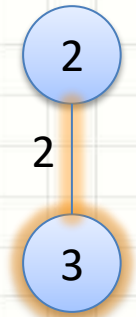
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O

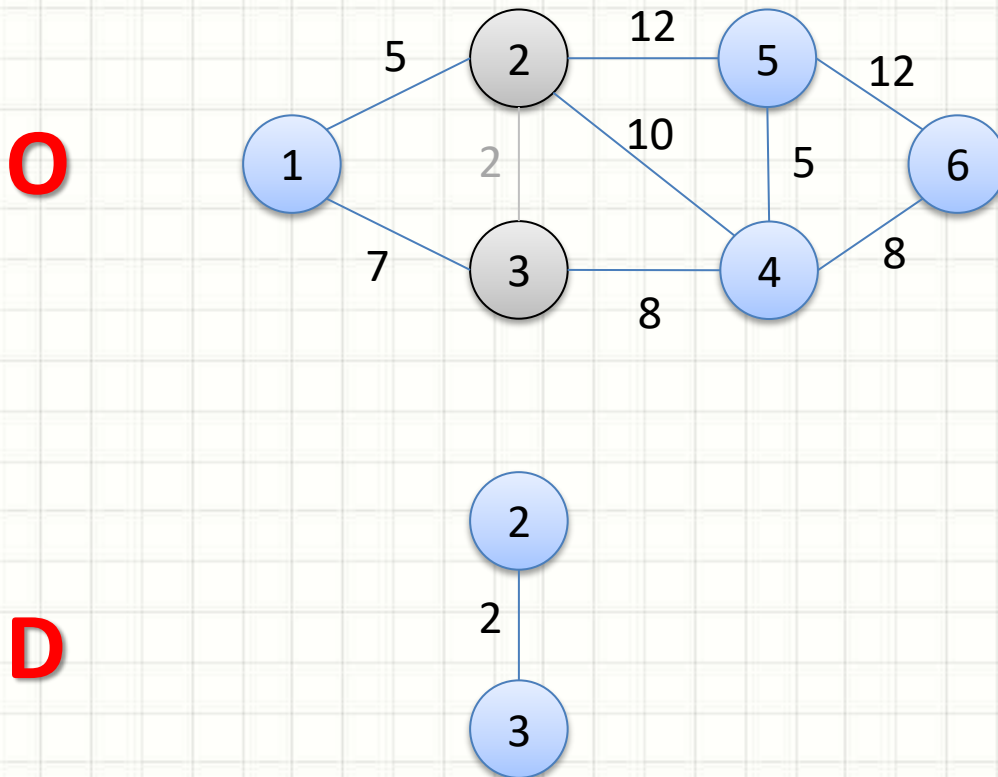


D



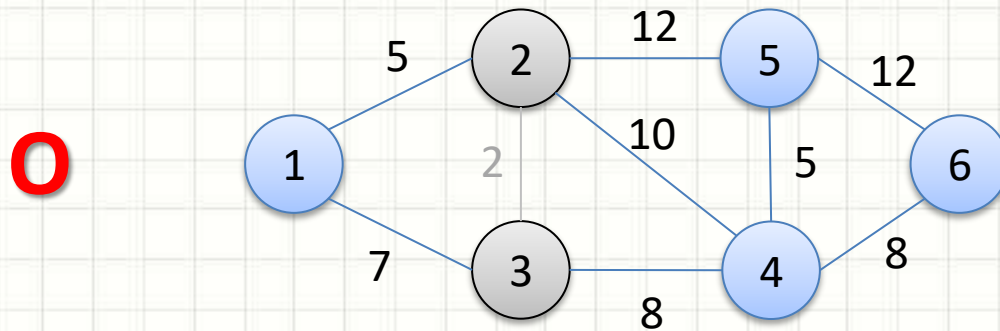
Aplicando o Algoritmo de Prim

5. Se ainda há nós em **O** que não estão em **D**, volte ao passo 3



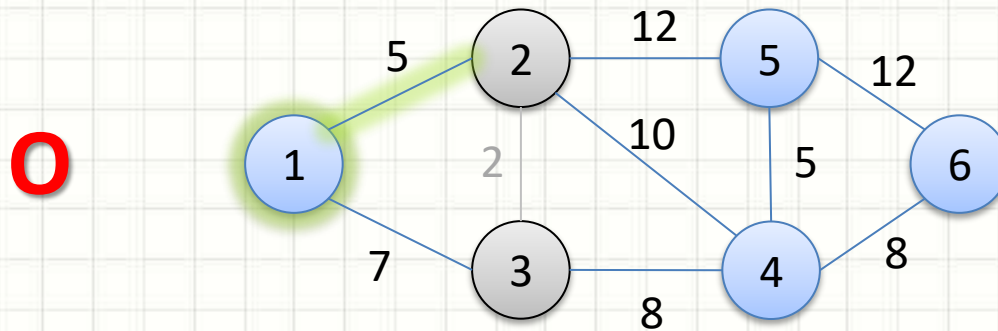
Aplicando o Algoritmo de Prim

3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

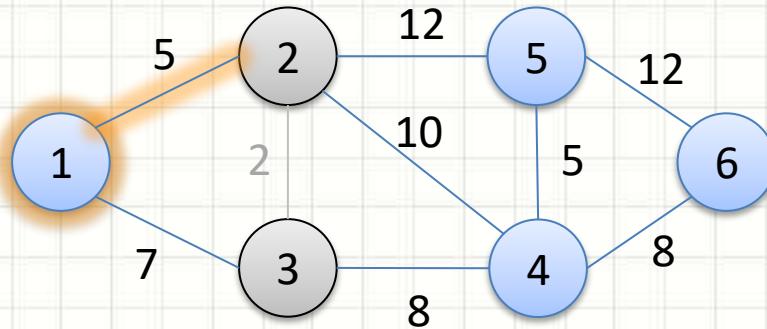
3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



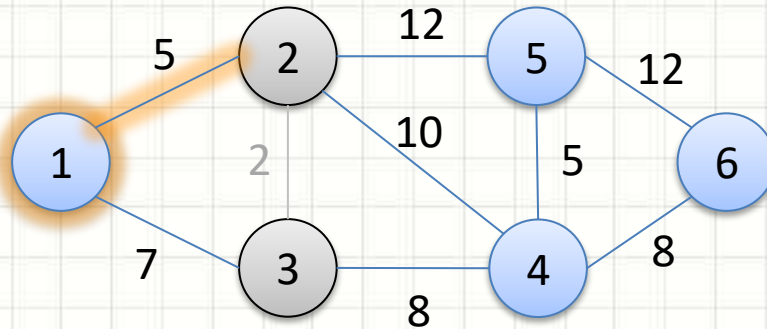
D



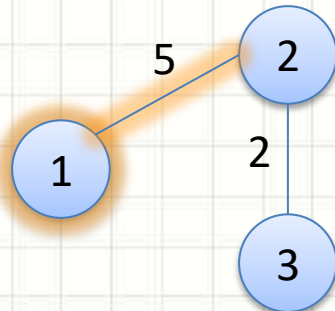
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O

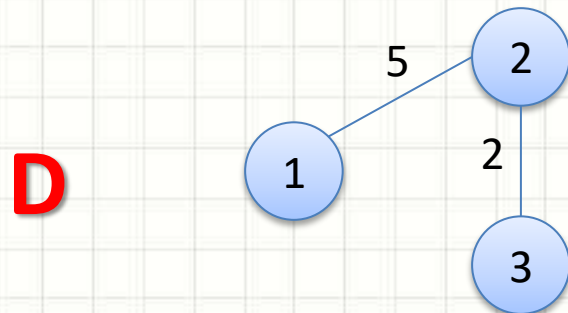
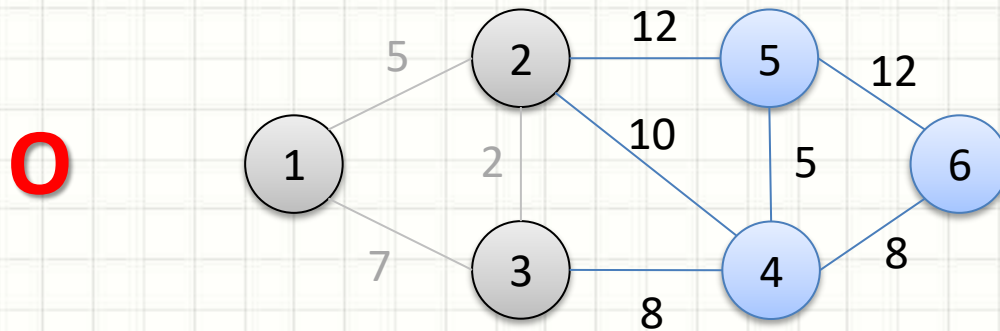


D



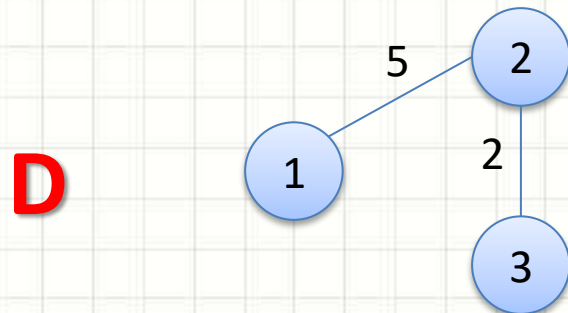
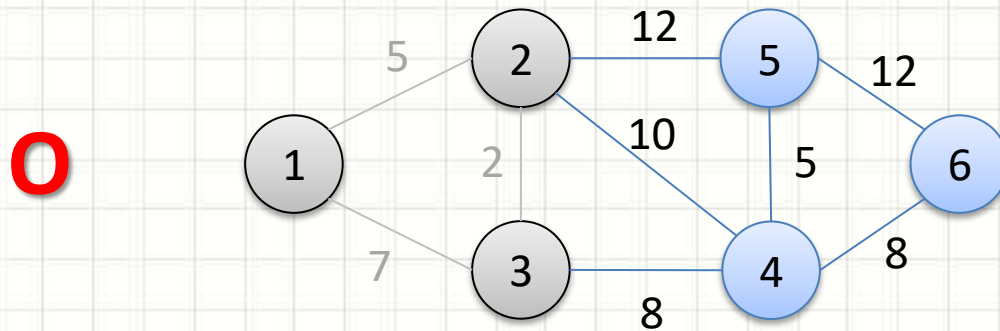
Aplicando o Algoritmo de Prim

5. Se ainda há nós em **O** que não estão em **D**, volte ao passo 3



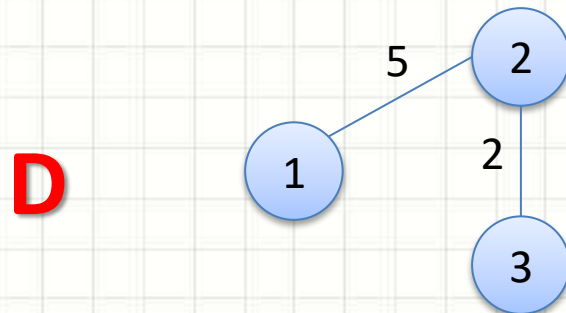
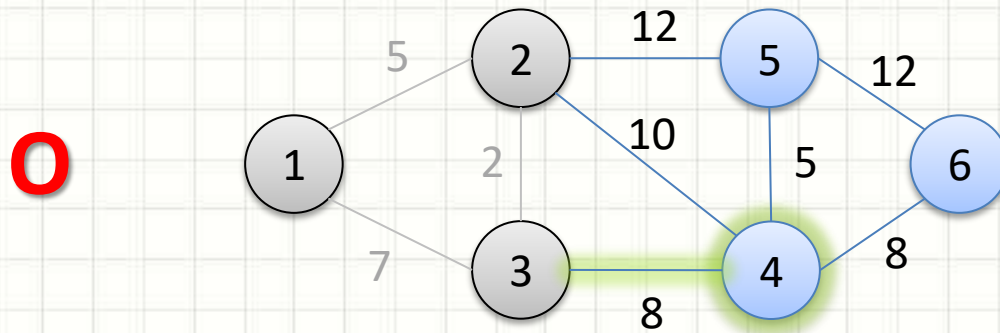
Aplicando o Algoritmo de Prim

3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

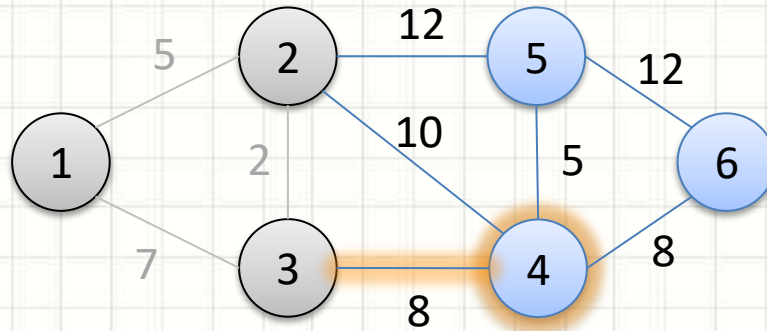
3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



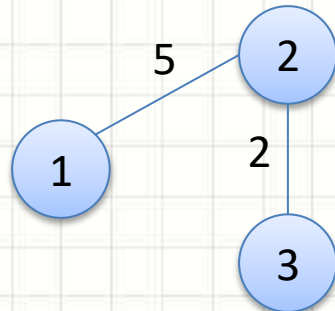
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



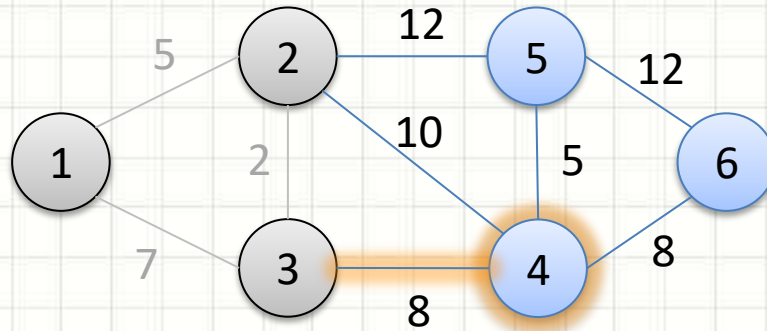
D



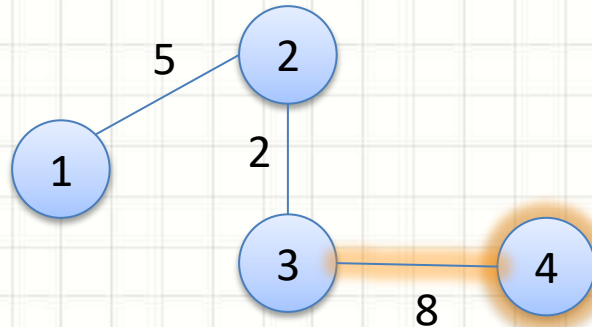
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



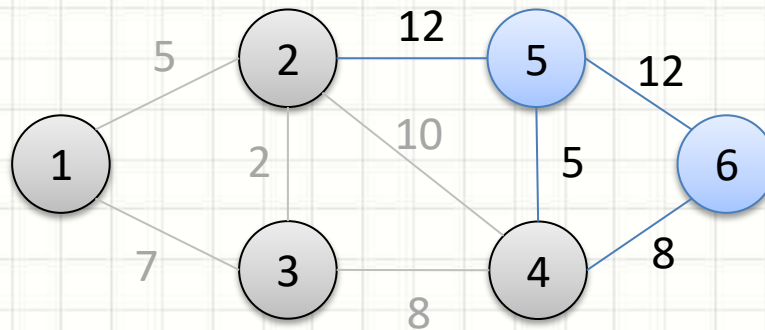
D



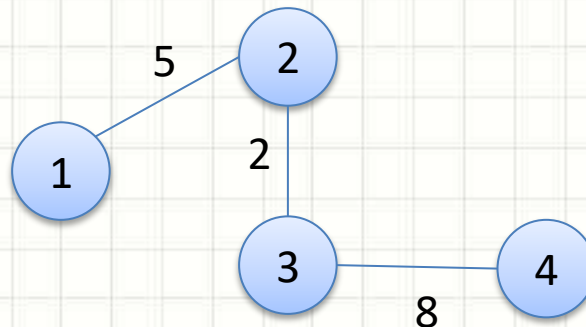
Aplicando o Algoritmo de Prim

5. Se ainda há nós em **O** que não estão em **D**, volte ao passo 3

O

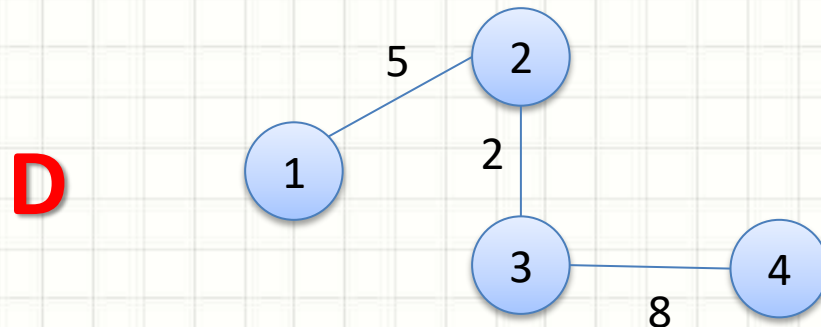
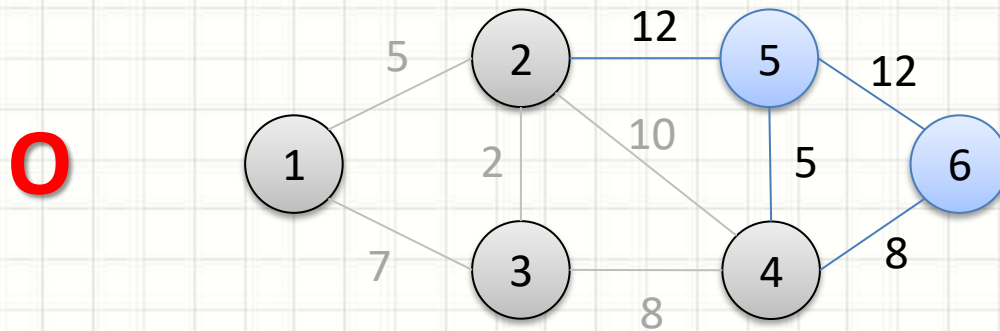


D



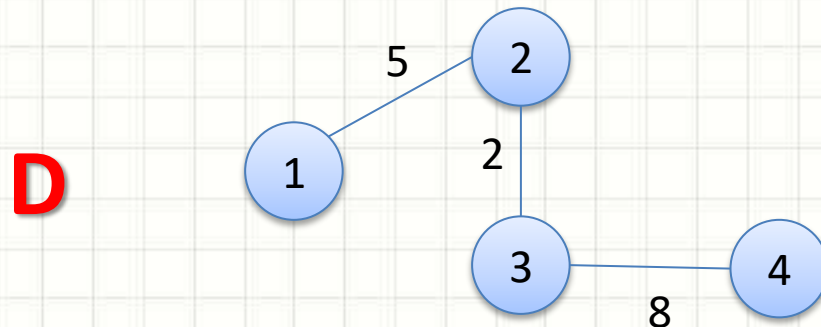
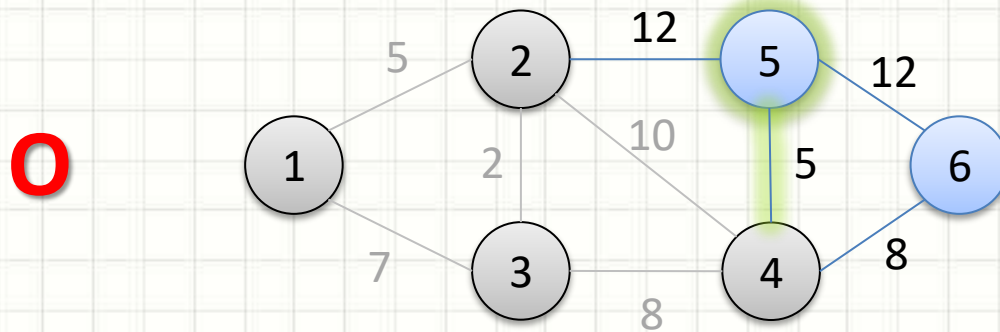
Aplicando o Algoritmo de Prim

3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

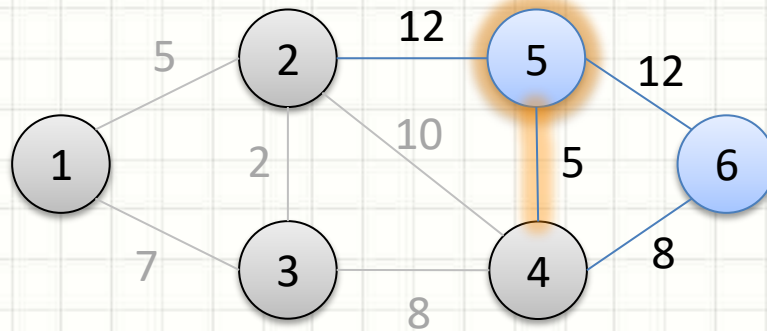
3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



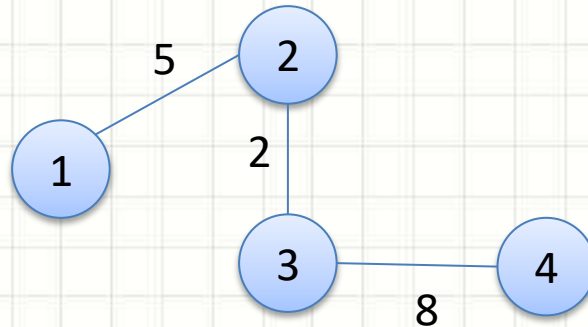
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



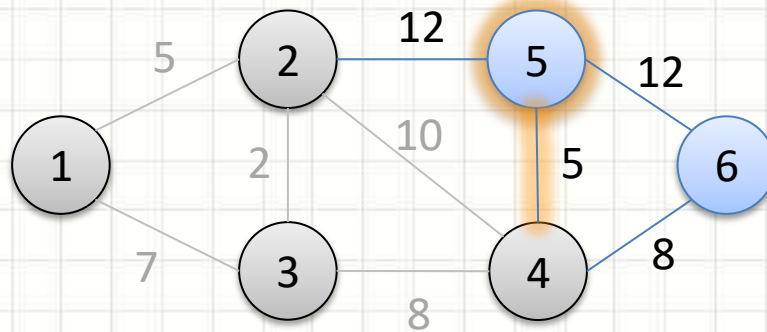
D



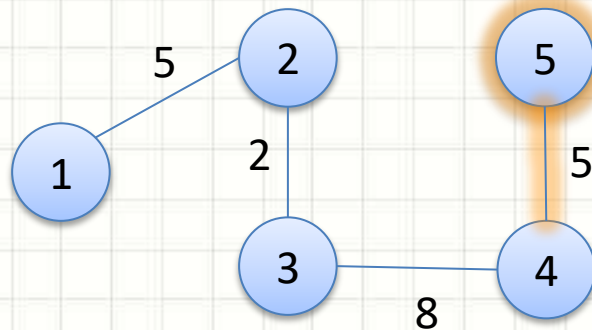
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



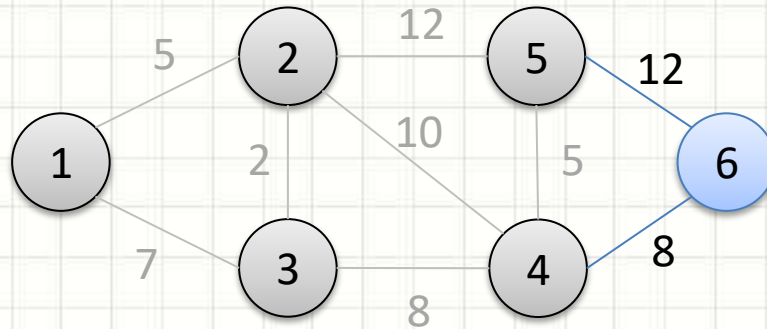
D



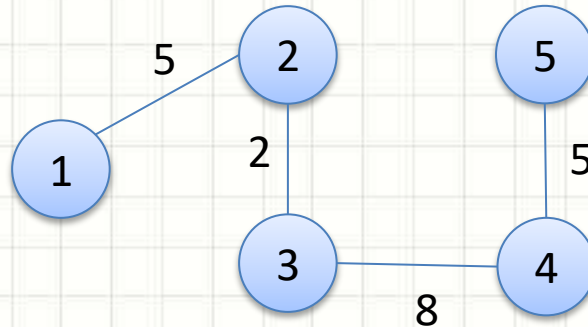
Aplicando o Algoritmo de Prim

5. Se ainda há nós em **O** que não estão em **D**, volte ao passo 3

O

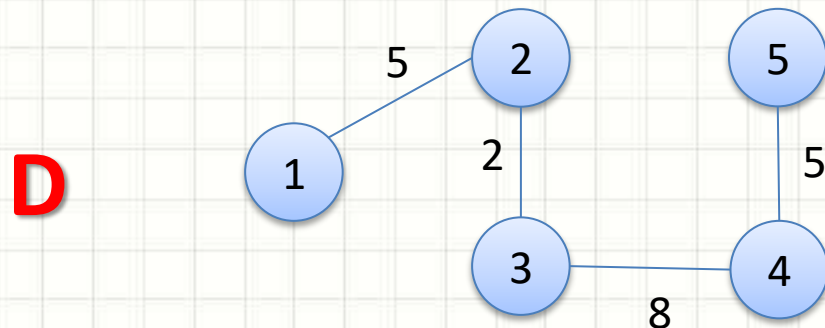
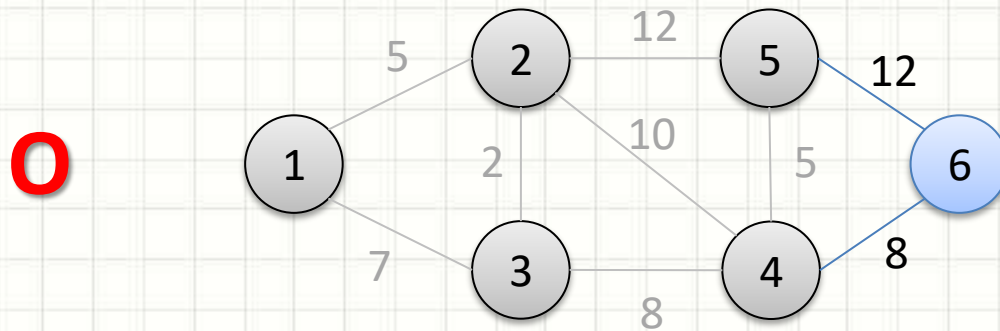


D



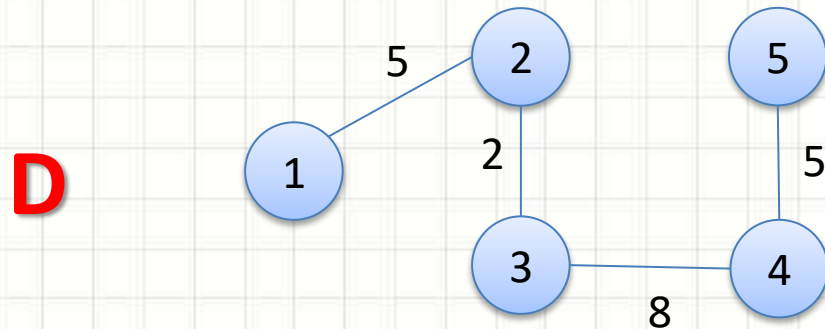
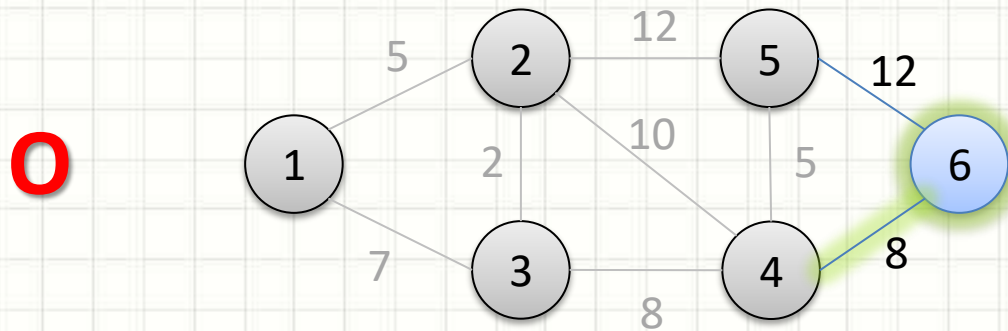
Aplicando o Algoritmo de Prim

3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



Aplicando o Algoritmo de Prim

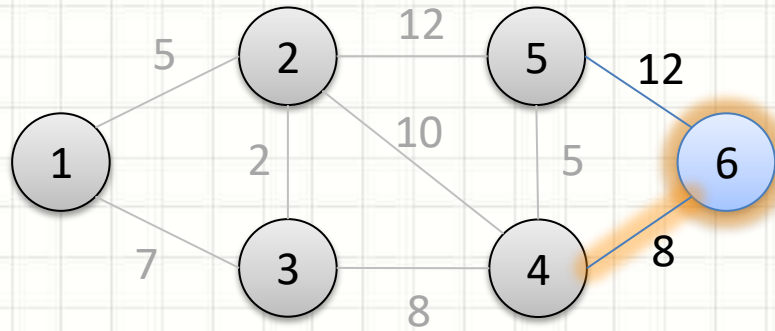
3. Em **O**, encontre o arco **L** de menor custo que está ligado a qualquer **nó** de **D** por **L** e seu nó **B**



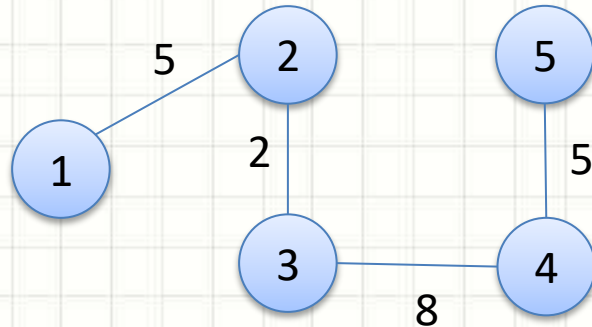
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O



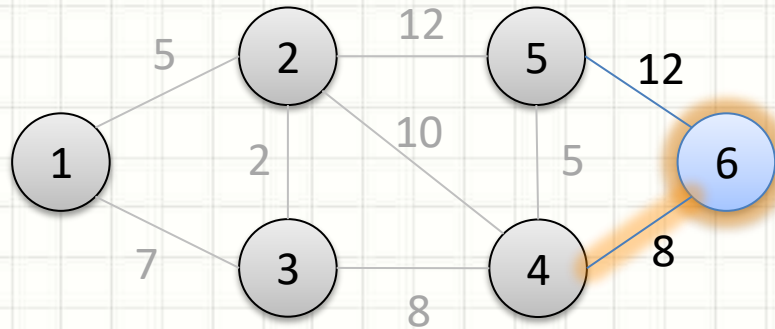
D



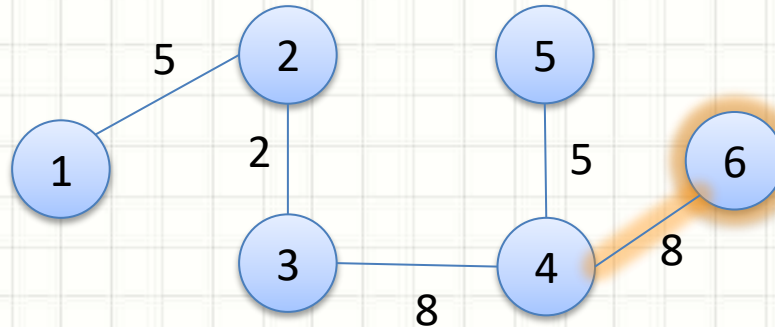
Aplicando o Algoritmo de Prim

4. Transfira **B** e **L** para **D**

O

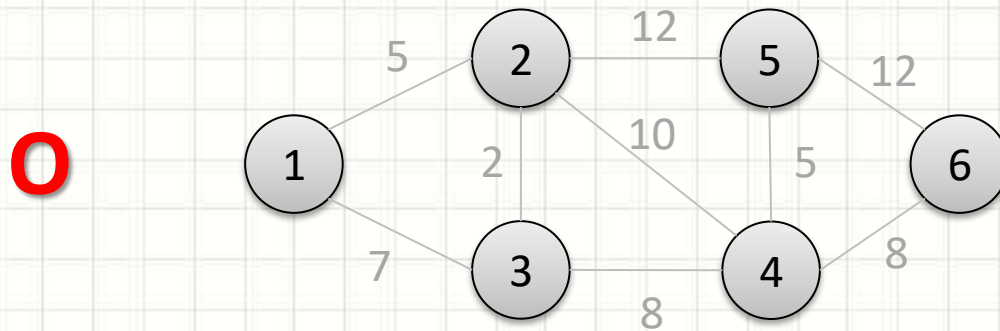


D

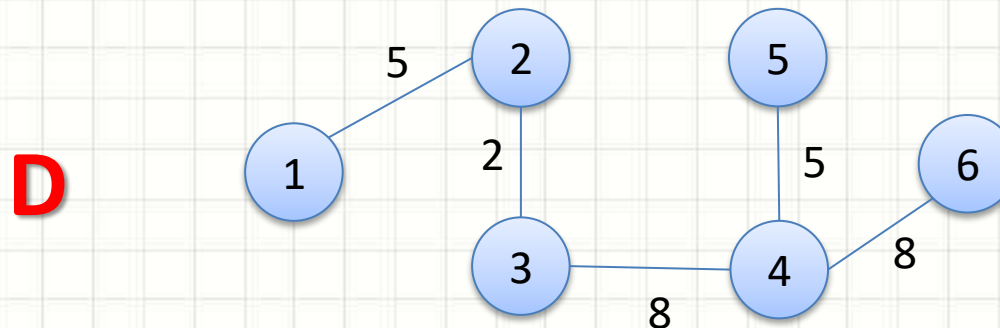


Aplicando o Algoritmo de Prim

5. Se ainda há nós em **O** que não estão em **D**, volte ao passo 3



Acabou!



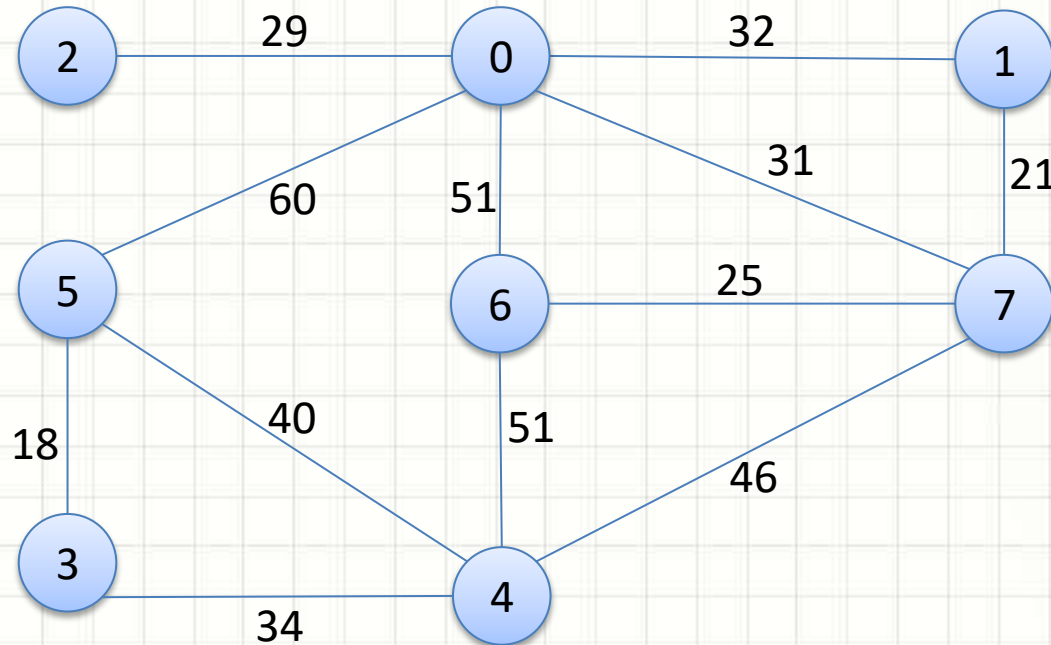
Custo: 28



EXERCÍCIOS

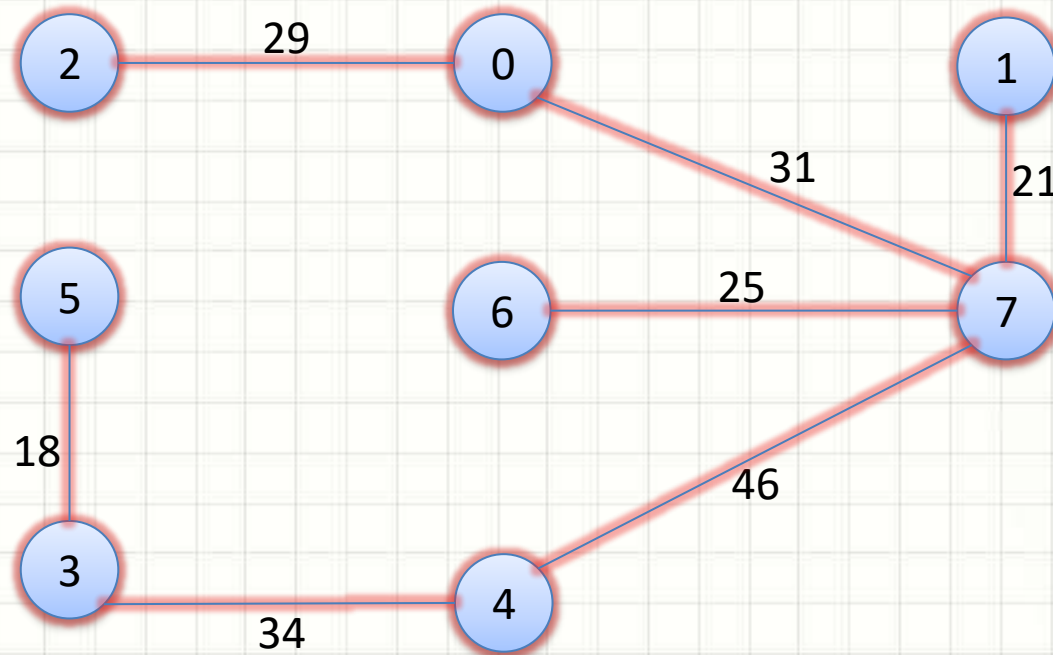
Exercício

1. Aplique o Algoritmo de Prim



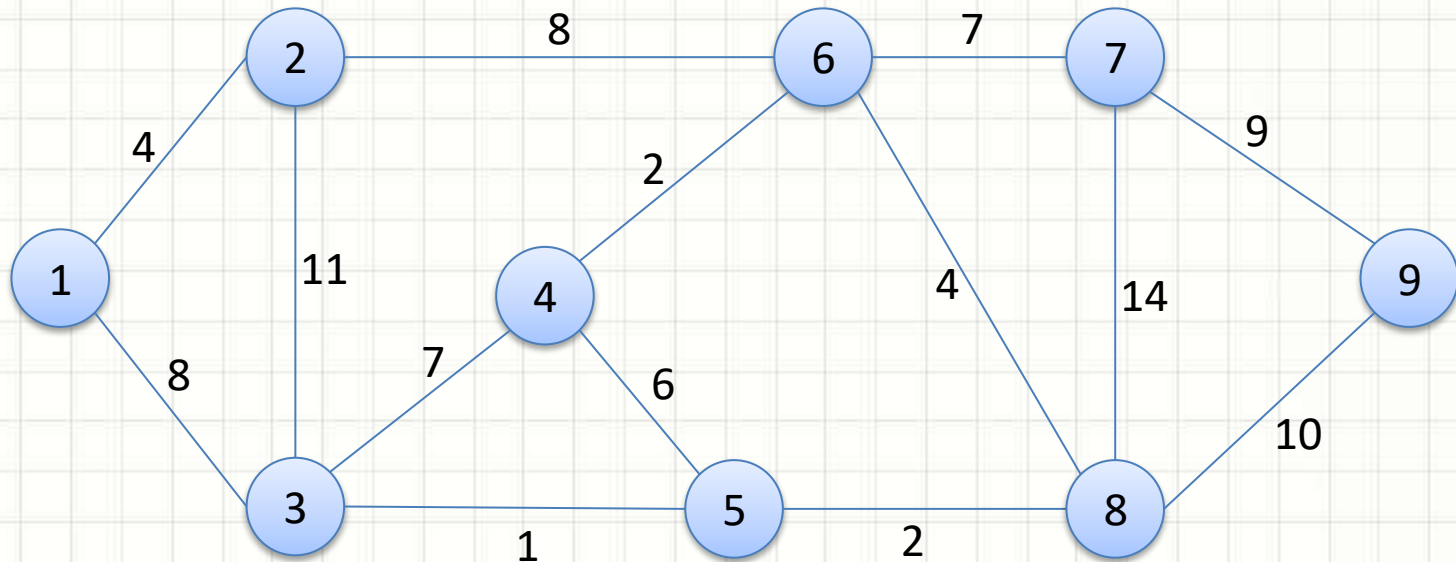
Exercício

1. Aplique o Algoritmo de Prim



Exercício (para entrega!)

2. Aplique o Algoritmo de Prim





CONCLUSÕES

Resumo

- Árvores, Árvores Geradoras...
 - Árvores Geradoras Mínimas
 - Algoritmo de Prim
 - **TAREFA:** Exercícios Aula 2
-

- Há outras formas de resolver?
 - Algoritmo de Kruskal



PERGUNTAS?