

# ALGORITMOS EM GRAFOS

## ÁRVORES DE STEINER

Prof. Alexei Machado

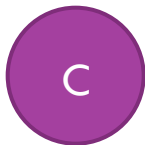
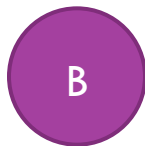
## Referência principal

GOLDBARG, Marco Cesar; GOLDBARG, Elizabeth. **Grafos**: conceitos, algoritmos e aplicações. Rio de Janeiro, RJ: Elsevier, Campus, c2012. xv, 622 p. ISBN 9788535257168.

# Problema de Steiner

3

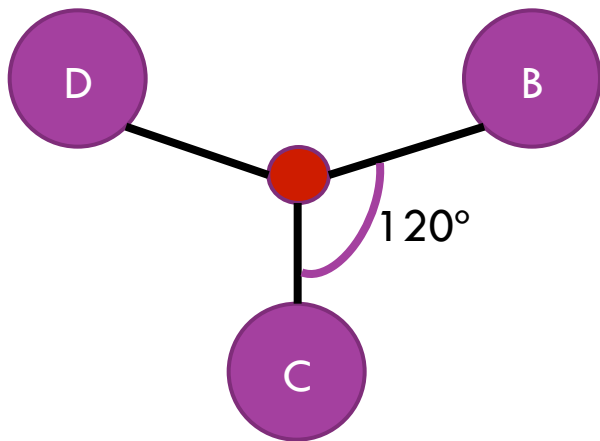
- Problema original: dados três pontos no plano euclidiano, encontre um quarto ponto tal que a soma das distâncias aos pontos originais seja mínima



# Problema de Steiner

4

- Problema original: dados três pontos no plano euclidiano, encontre um quarto ponto tal que a soma das distâncias aos pontos originais seja mínima



# Problema de Steiner em grafos

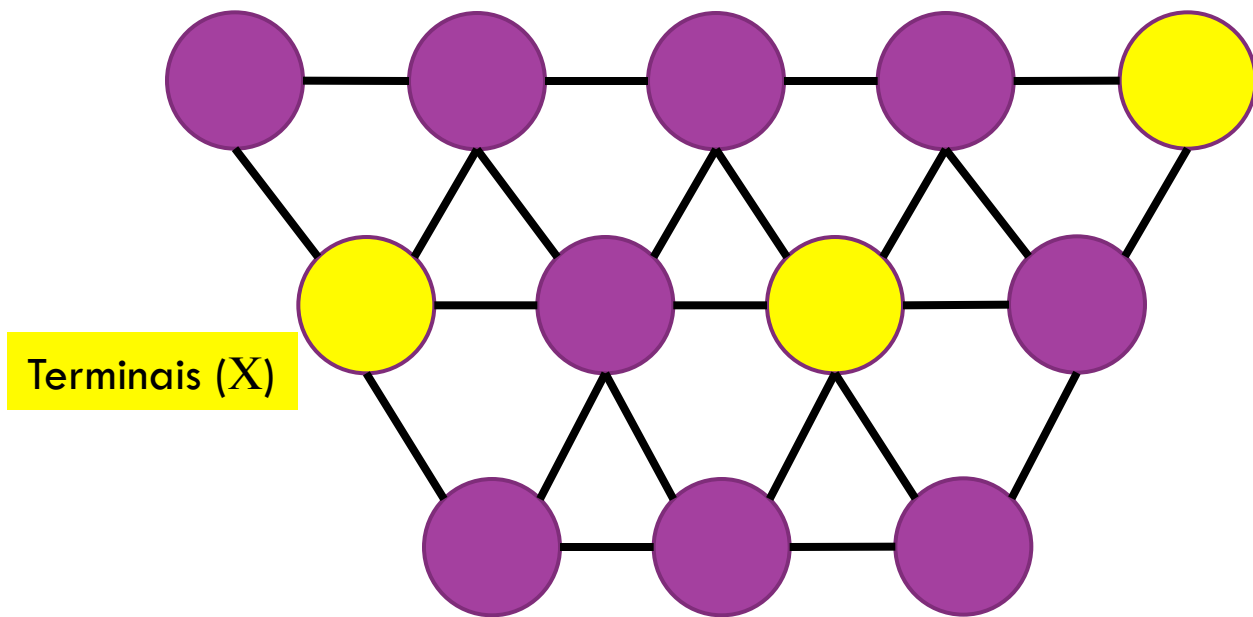
5

- Conectar, a um custo mínimo, um conjunto de **vértices obrigatórios** (terminais) em um grafo  $G$

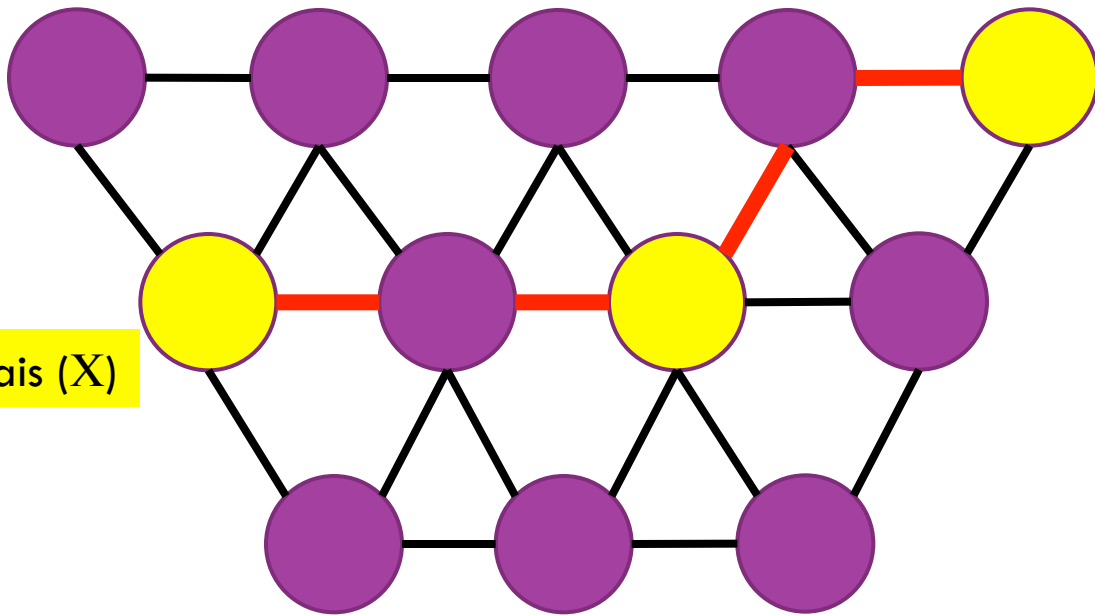
# Árvore de Steiner

6

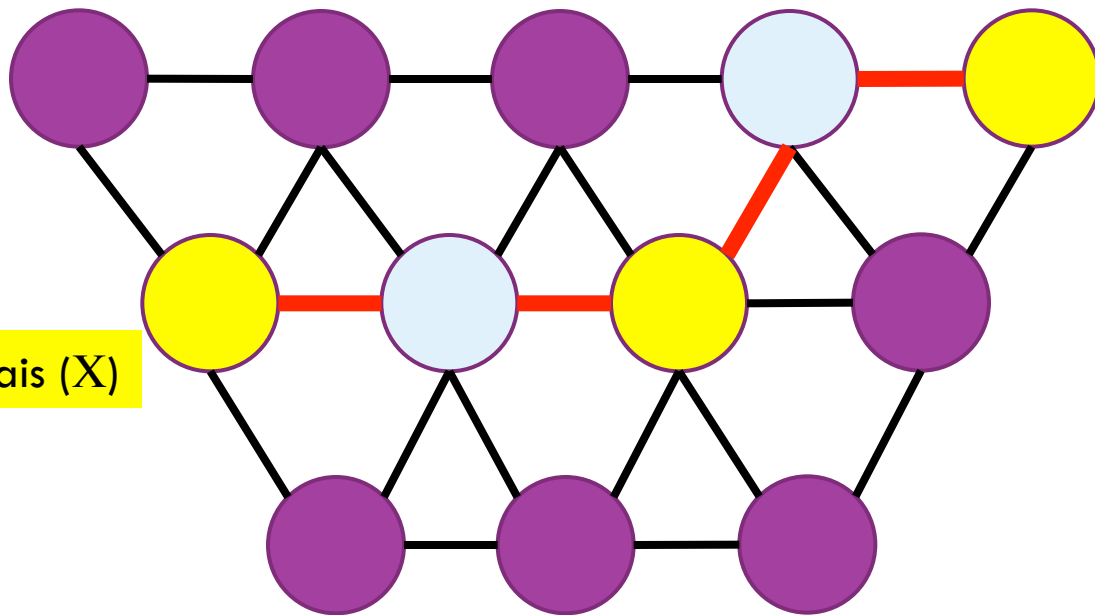
- Árvore que possui o conjunto  $X$  de terminais e outros vértices que possam ajudar na ligação dos vértices de  $X$ .
- Estes são os **vértices de Steiner**, e seu conjunto é chamado  $\Lambda$



Terminais (X)

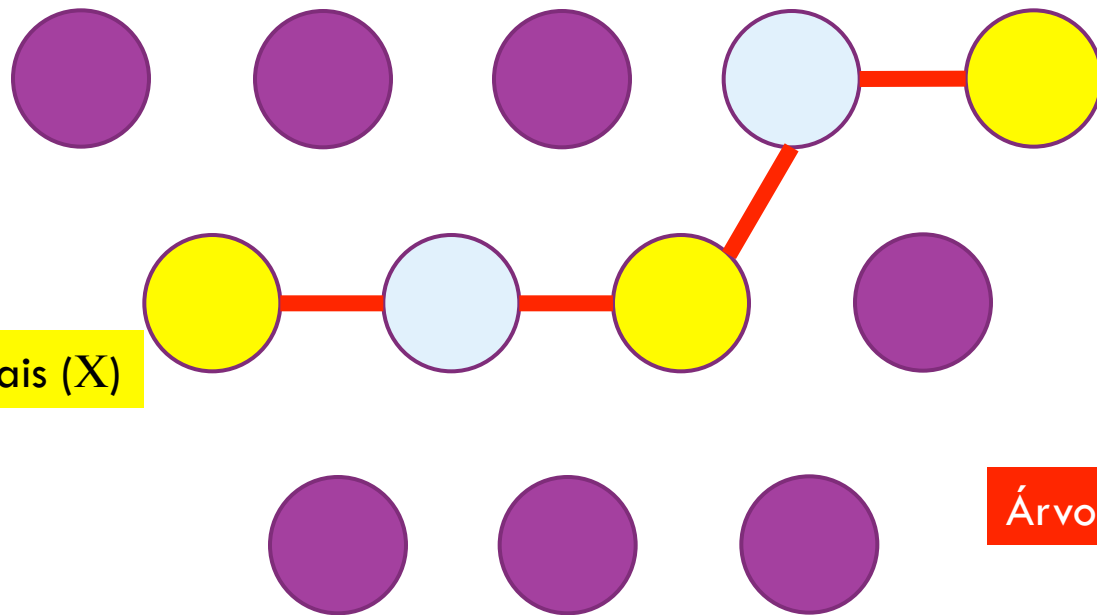






Terminais (X)

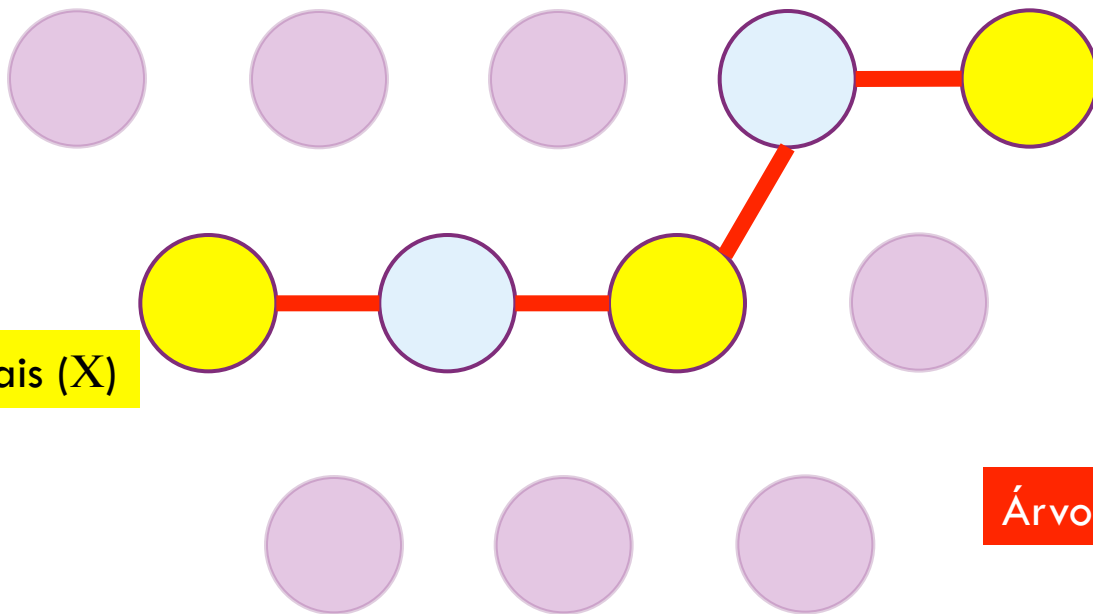
Vértices de Steiner( $\Lambda$ )



Terminais ( $X$ )

Vértices de Steiner( $\Lambda$ )

Árvore de Steiner



Terminais (X)

Vértices de Steiner( $\Lambda$ )

Árvore de Steiner

# Árvores de Steiner

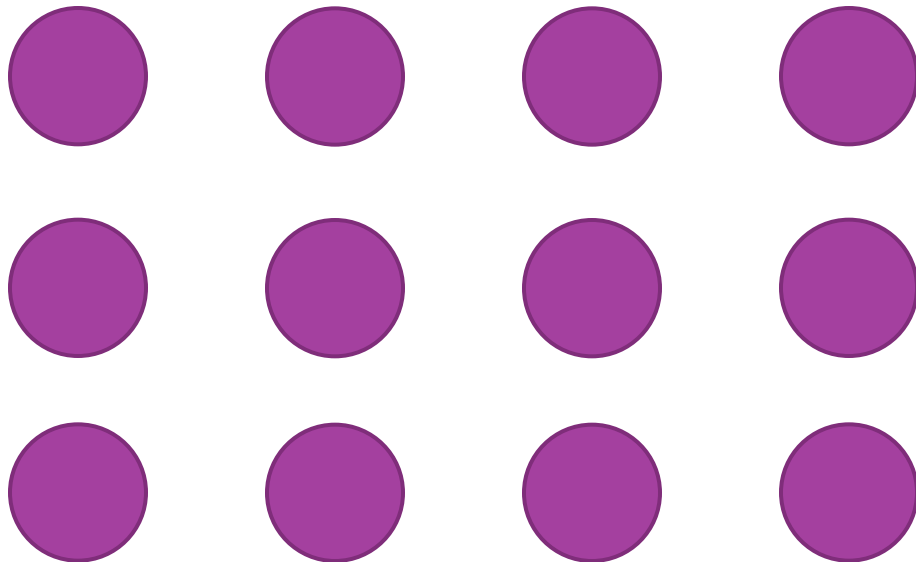
12

- Infelizmente, encontrar uma árvore de Steiner de custo mínimo é um problema NP-Difícil.

# Árvore de Steiner com norma linear

13

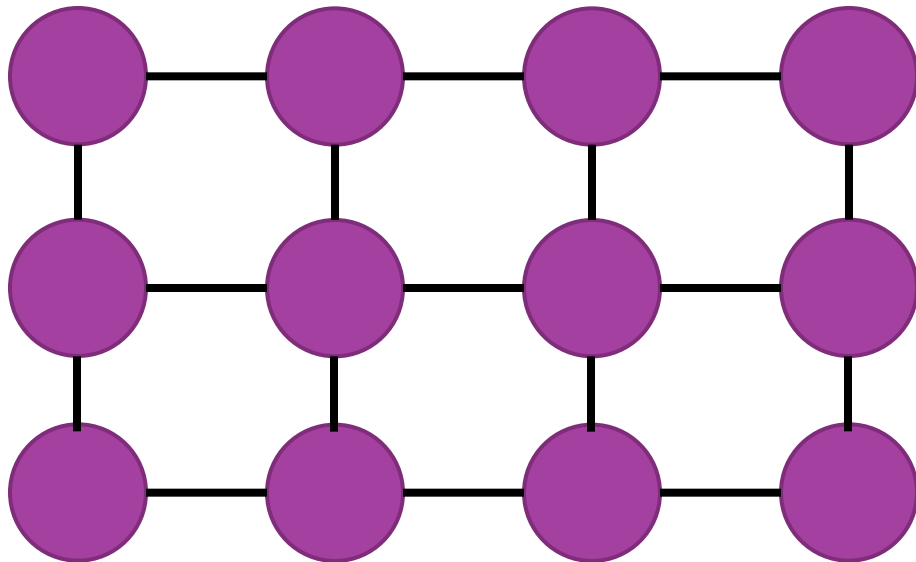
- Caso particular do problema de Steiner no plano euclidiano



# Árvore de Steiner com norma linear

14

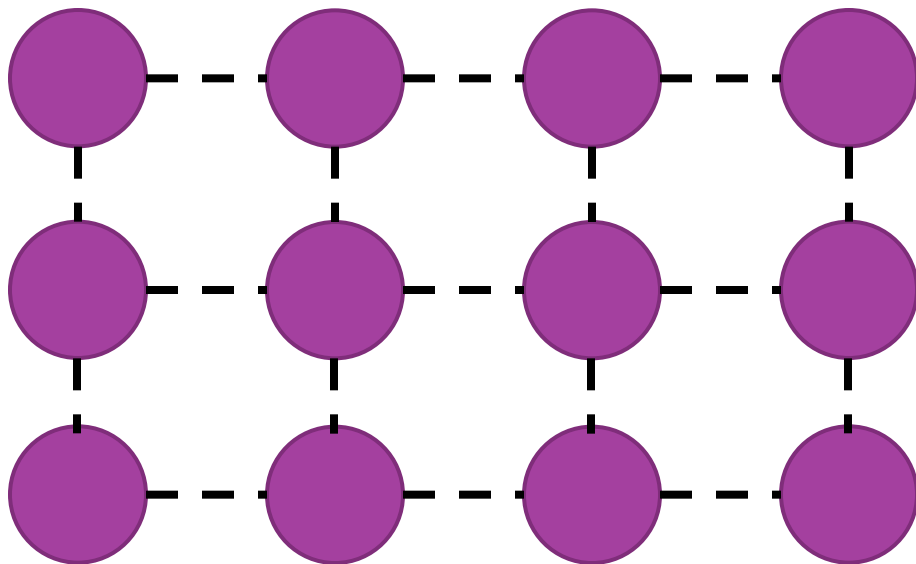
- Caso particular do problema de Steiner no plano euclidiano



# Árvore de Steiner com norma linear

15

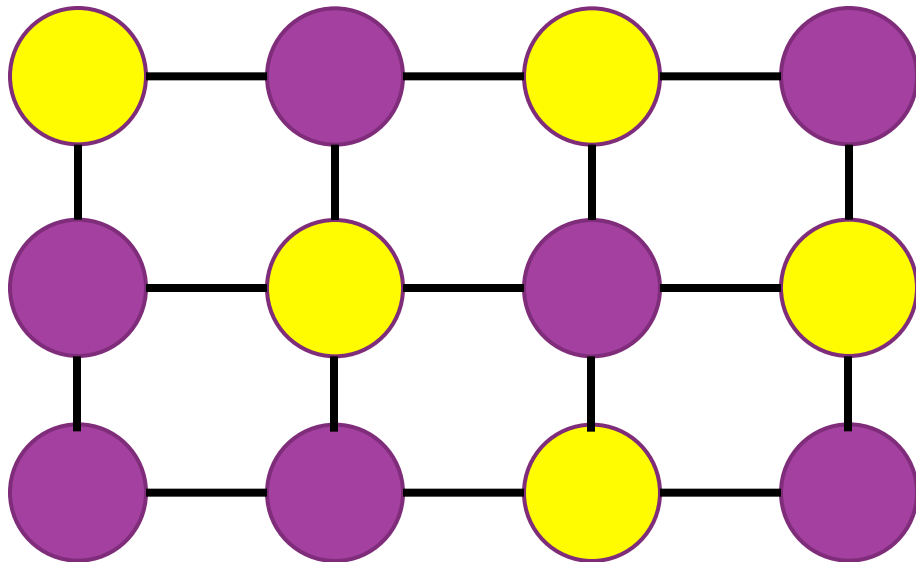
- Caso particular do problema de Steiner no plano euclidiano



# Árvore de Steiner com norma linear

16

- Caso particular do problema de Steiner no plano euclidiano

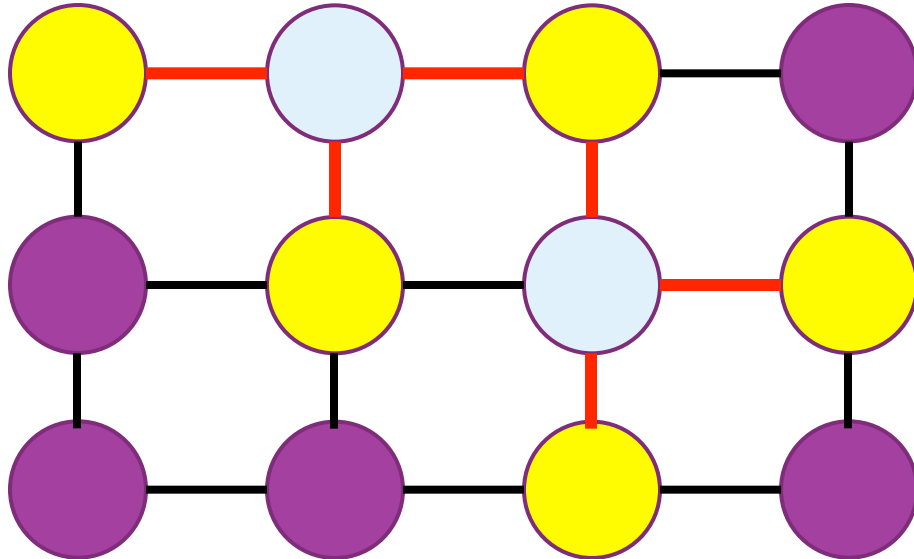




# Árvore de Steiner com norma linear

17

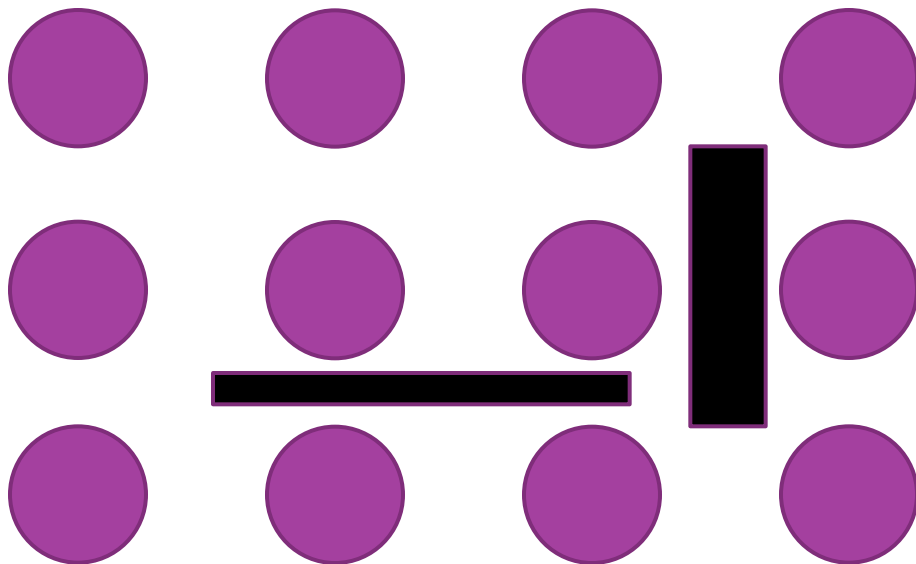
- Caso particular do problema de Steiner no plano euclidiano



# Árvore de Steiner com norma linear

18

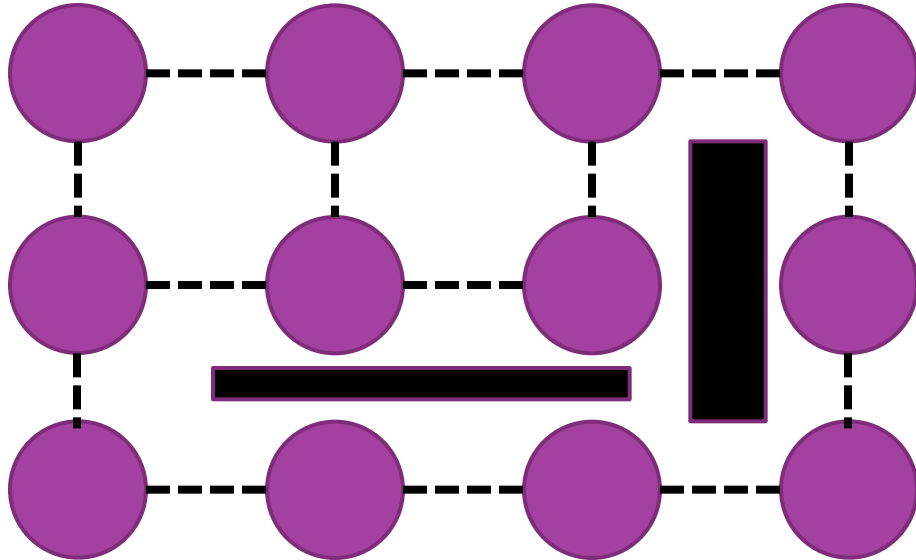
- Também há casos com *obstáculos*



# Árvore de Steiner com norma linear

19

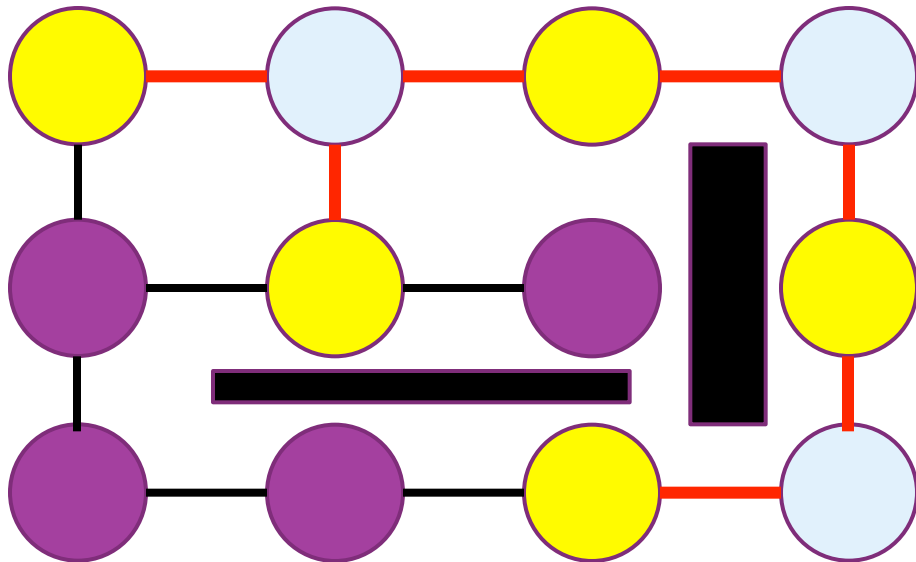
- Também há casos com *obstáculos*



# Árvore de Steiner com norma linear

20

- Também há casos com *obstáculos*



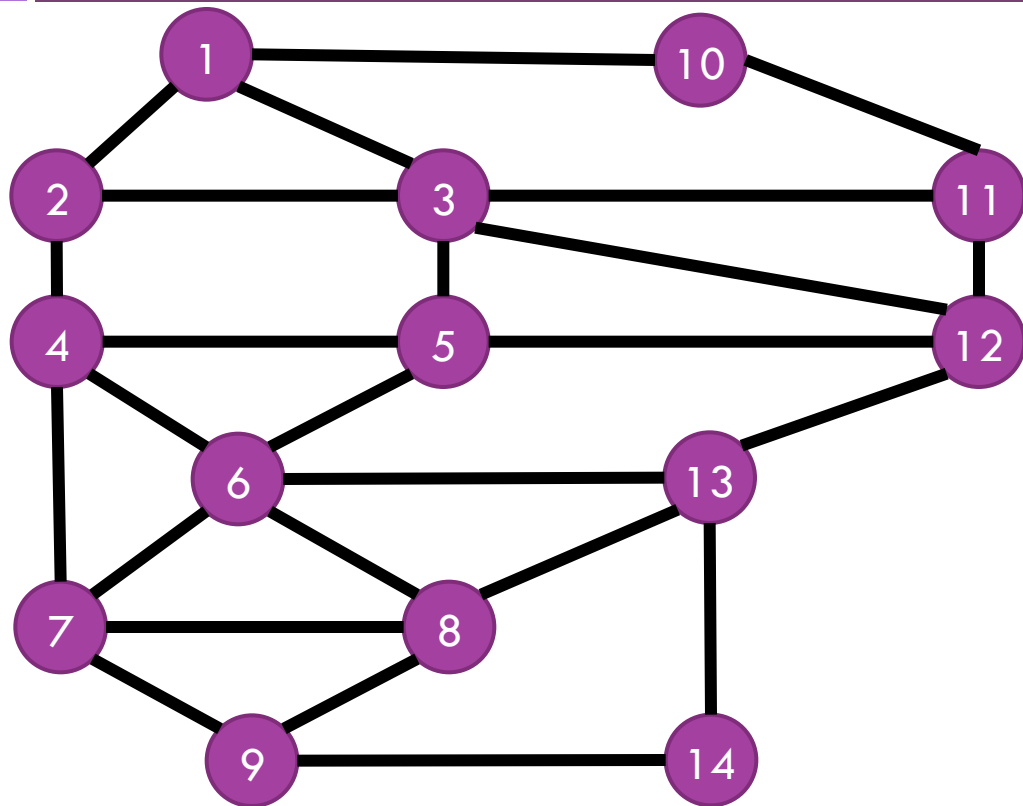
# Árvores de Steiner

21

- Em geral, geradas a partir de algoritmos heurísticos

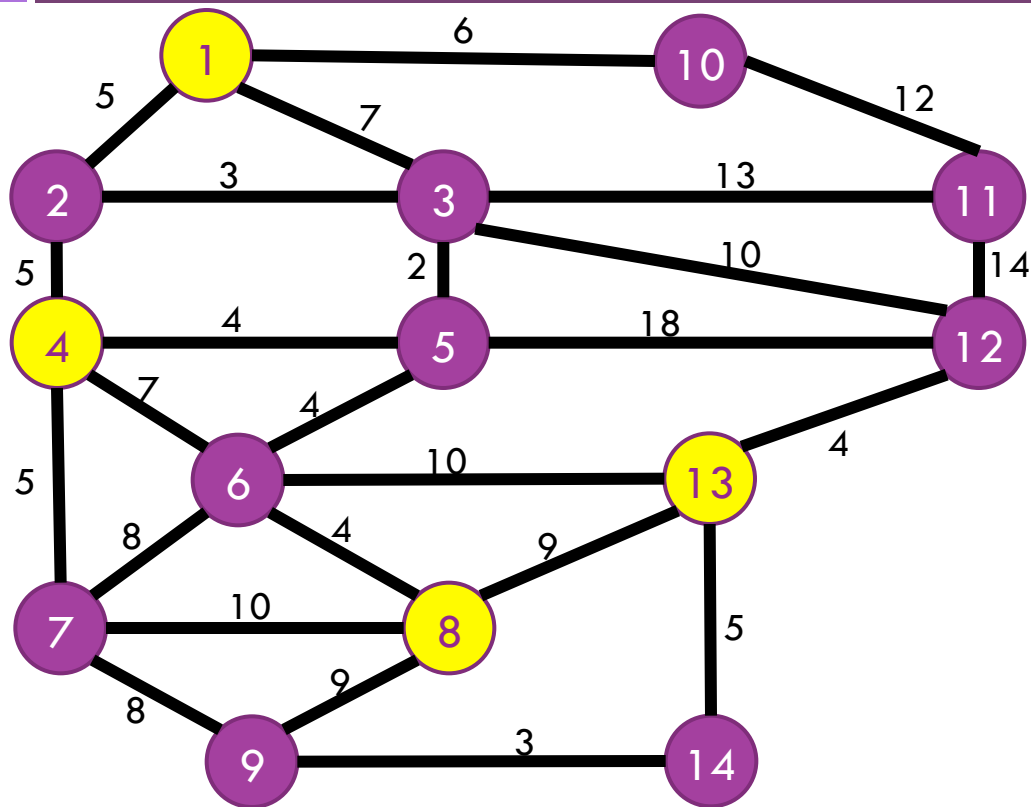
# Algoritmo da AGM com poda

22



# Algoritmo da AGM com poda

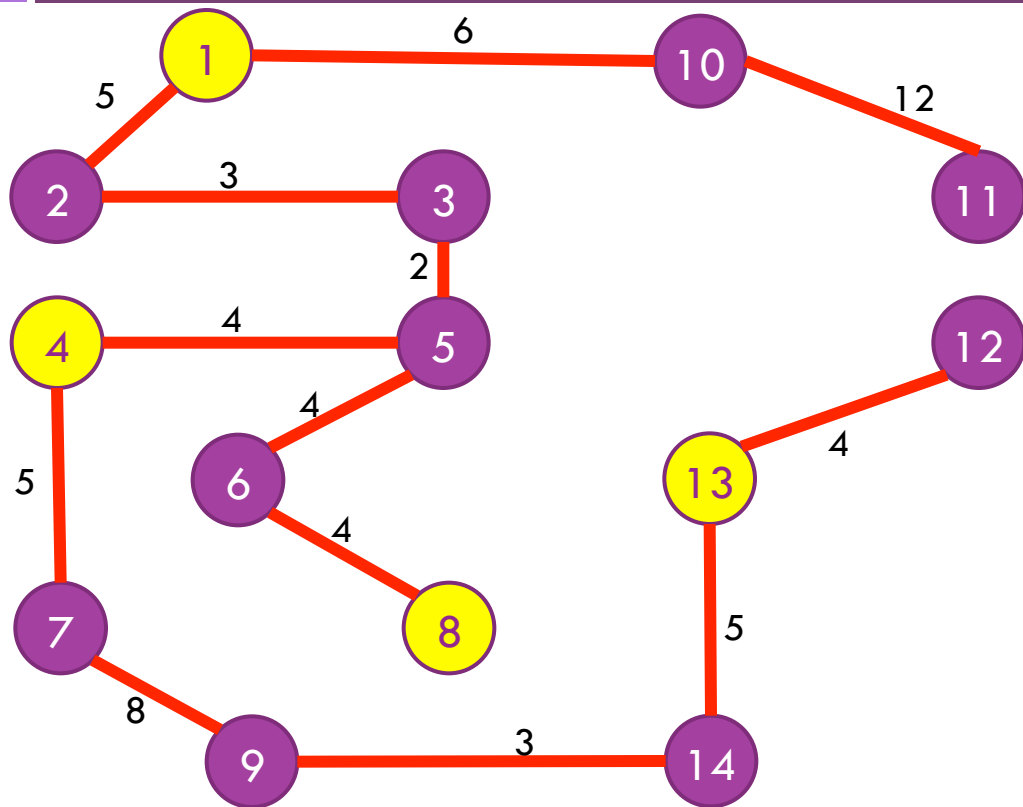
23



- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas

# Algoritmo da AGM com poda

24

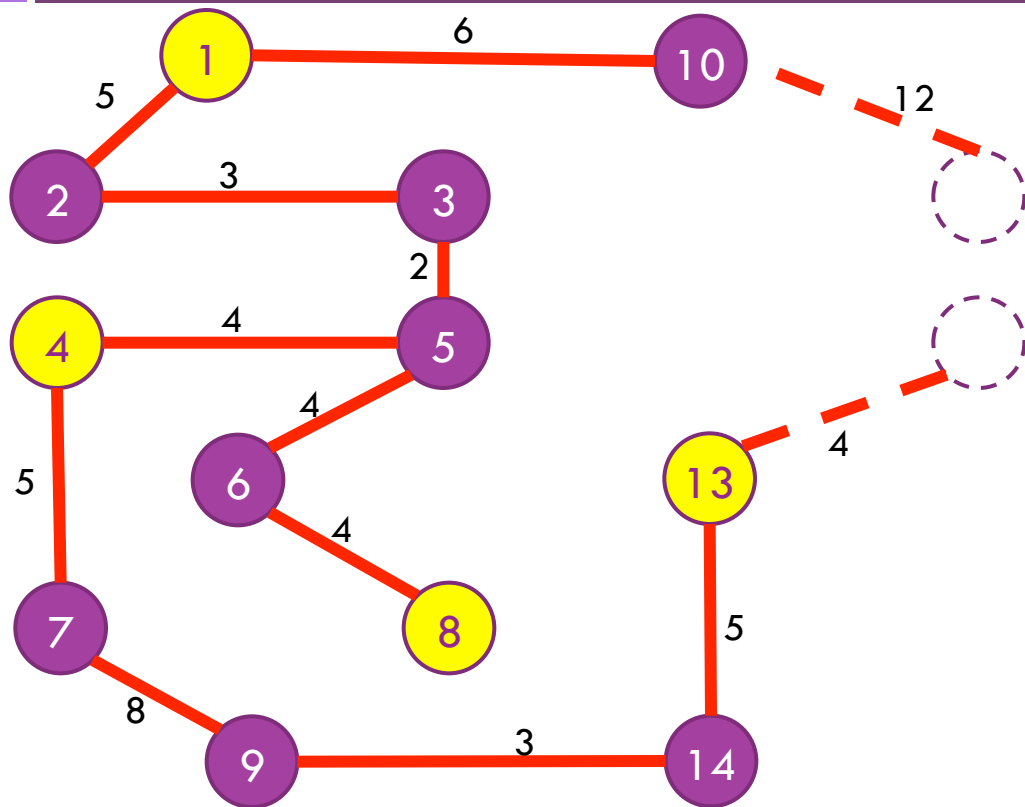


- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas



# Algoritmo da AGM com poda

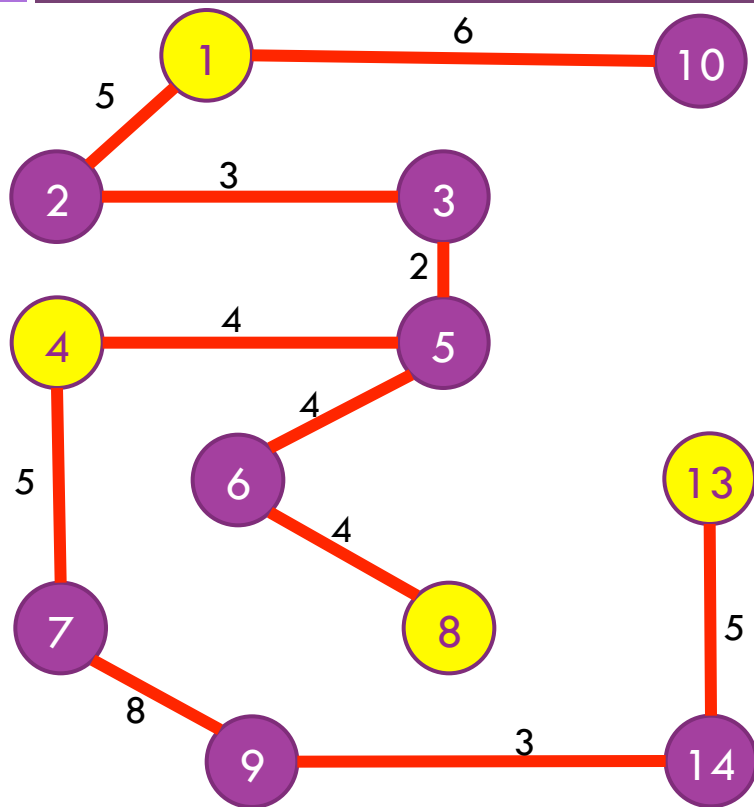
25



- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas

# Algoritmo da AGM com poda

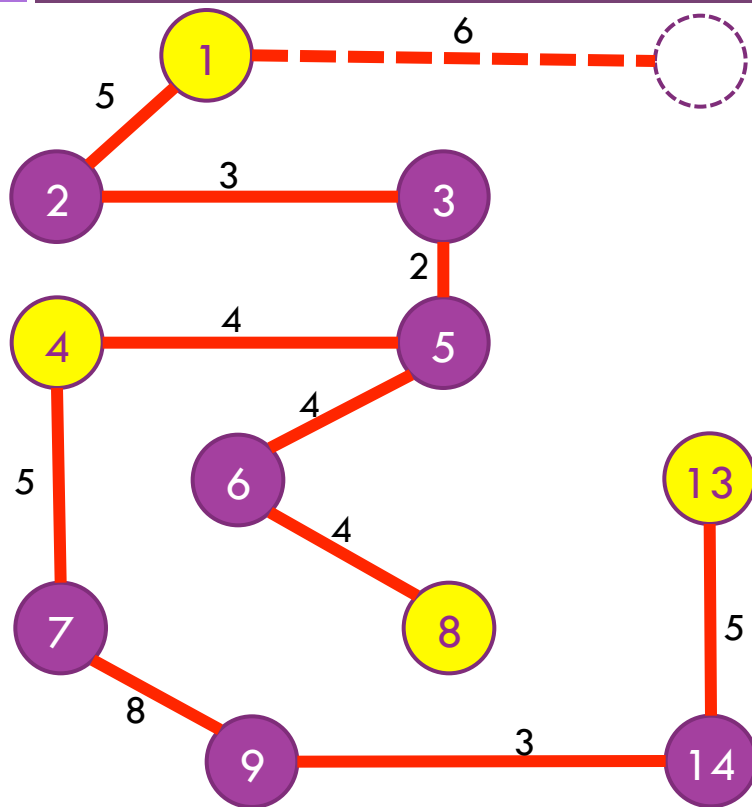
26



- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas

# Algoritmo da AGM com poda

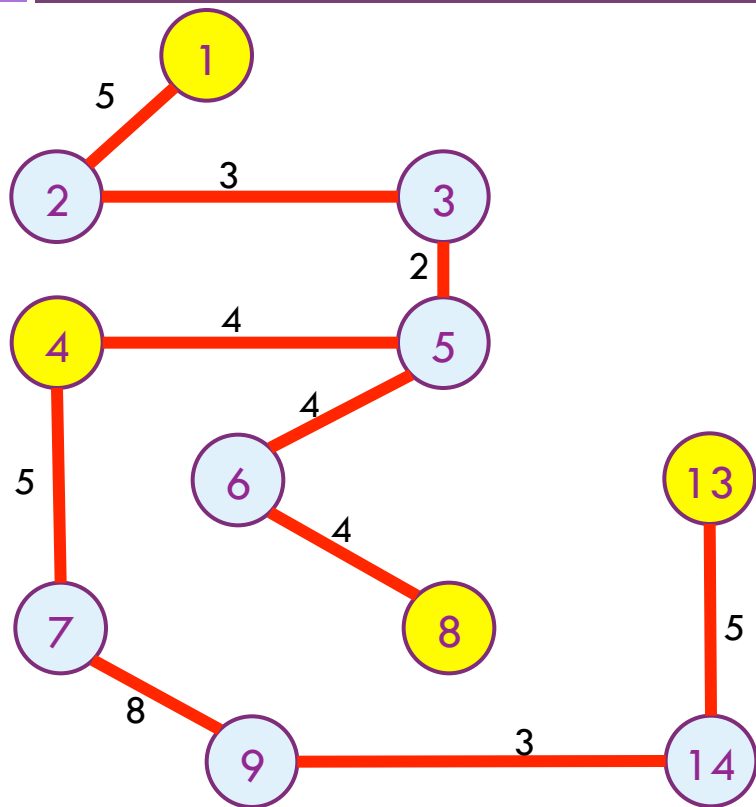
27



- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas

# Algoritmo da AGM com poda

28

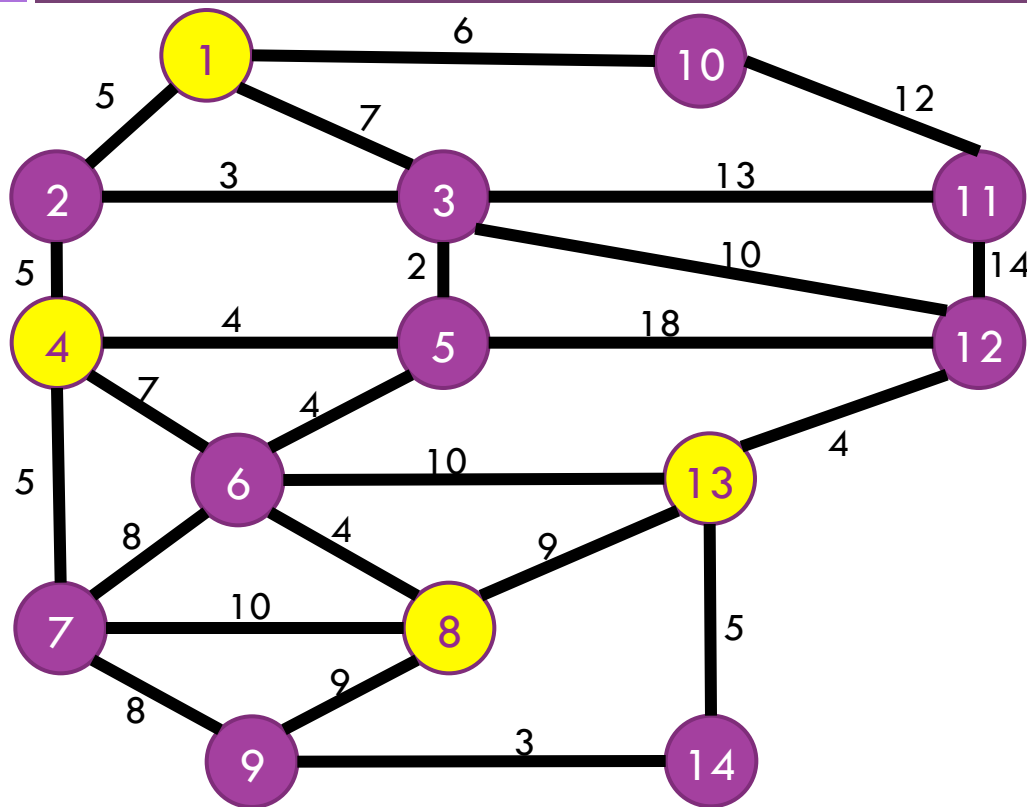


- 1) Determinar a AGM
- 2) Enquanto houver folhas que não são terminais, podar tais folhas

Árvore de Steiner

# Algoritmo KMB (Kou, Markowsky, Berman)

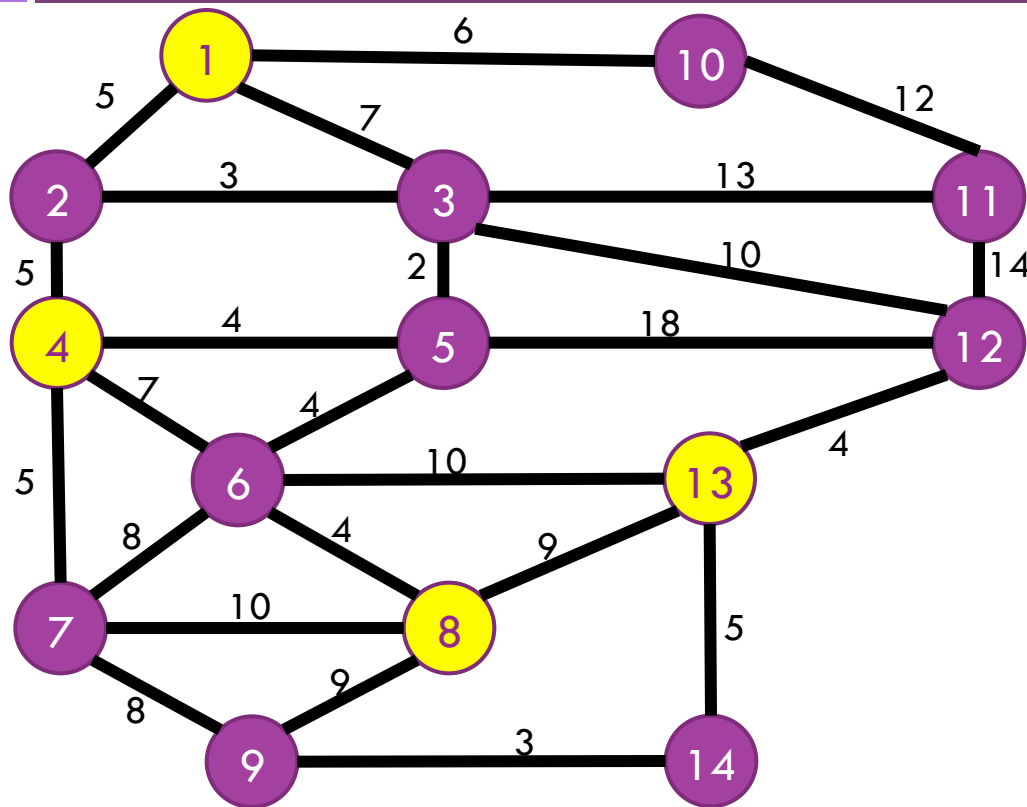
29



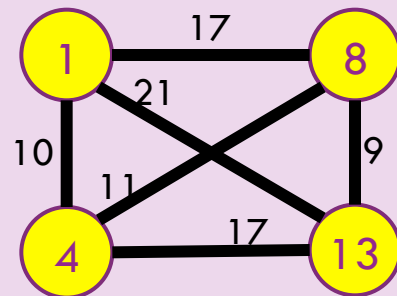
- 1) Determinar os caminhos mais curtos entre os vértices de  $X$
- 2) Criar um grafo  $G(S)$  completo com os vértices de  $X$  e as arestas dos caminhos mais curtos
- 3) Determinar a AGM de  $G(S)$
- 4) Substituir a AGM pelos caminhos reais

# Algoritmo KMB (Kou, Markowsky, Berman)

30

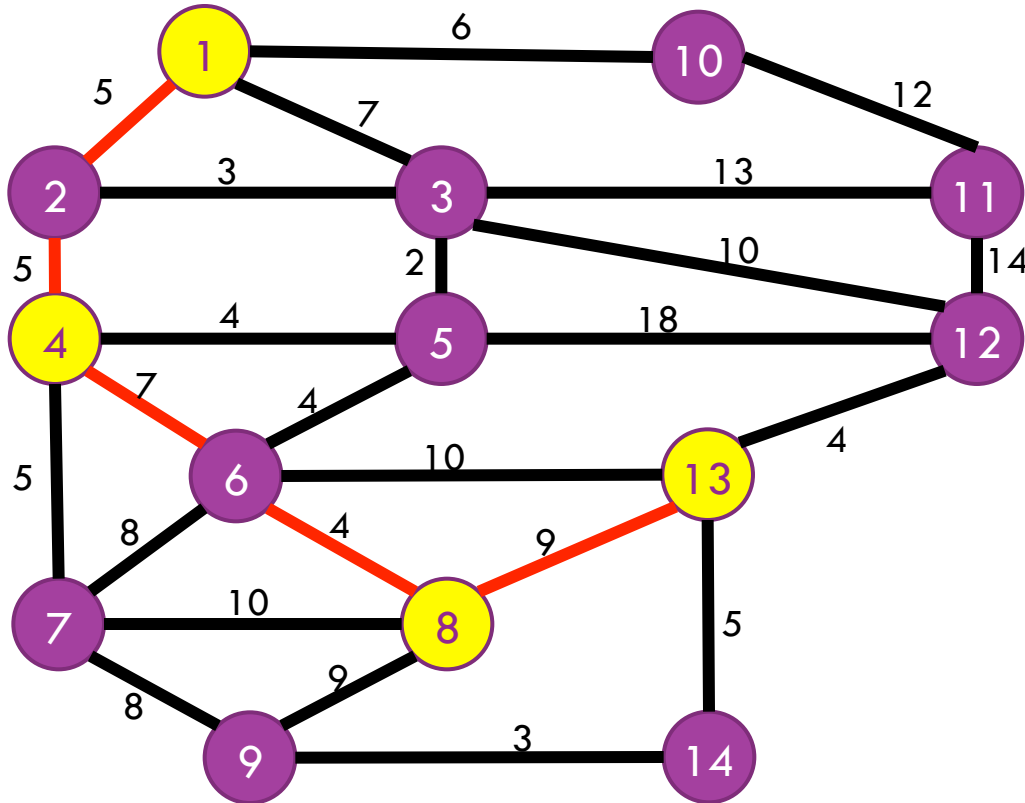


- 1) Determinar os caminhos mais curtos entre os vértices de  $X$
- 2) Criar um grafo  $G(S)$  completo com os vértices de  $X$  e as arestas dos caminhos mais curtos



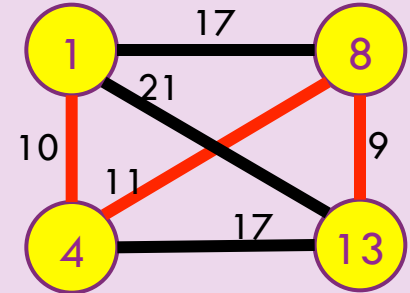
# Algoritmo KMB (Kou, Markowsky, Berman)

31



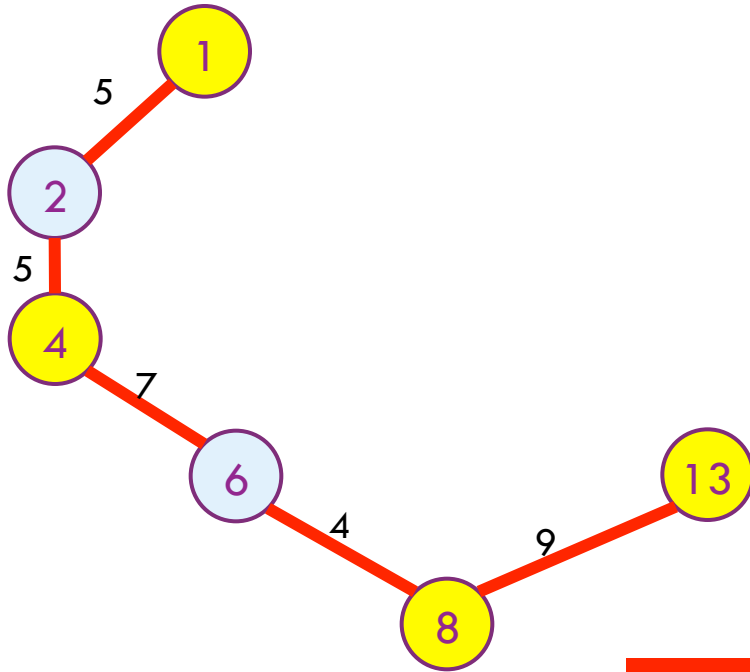
3) Determinar a AGM de  $G(S)$

4) Substituir a AGM pelos caminhos reais



# Algoritmo KMB (Kou, Markowsky, Berman)

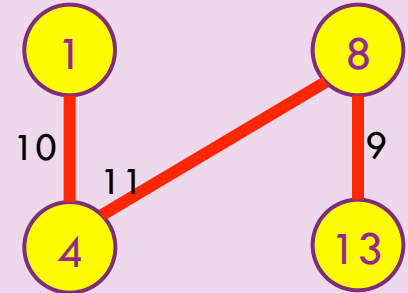
32



Árvore de Steiner

3) Determinar a AGM de  $G(S)$

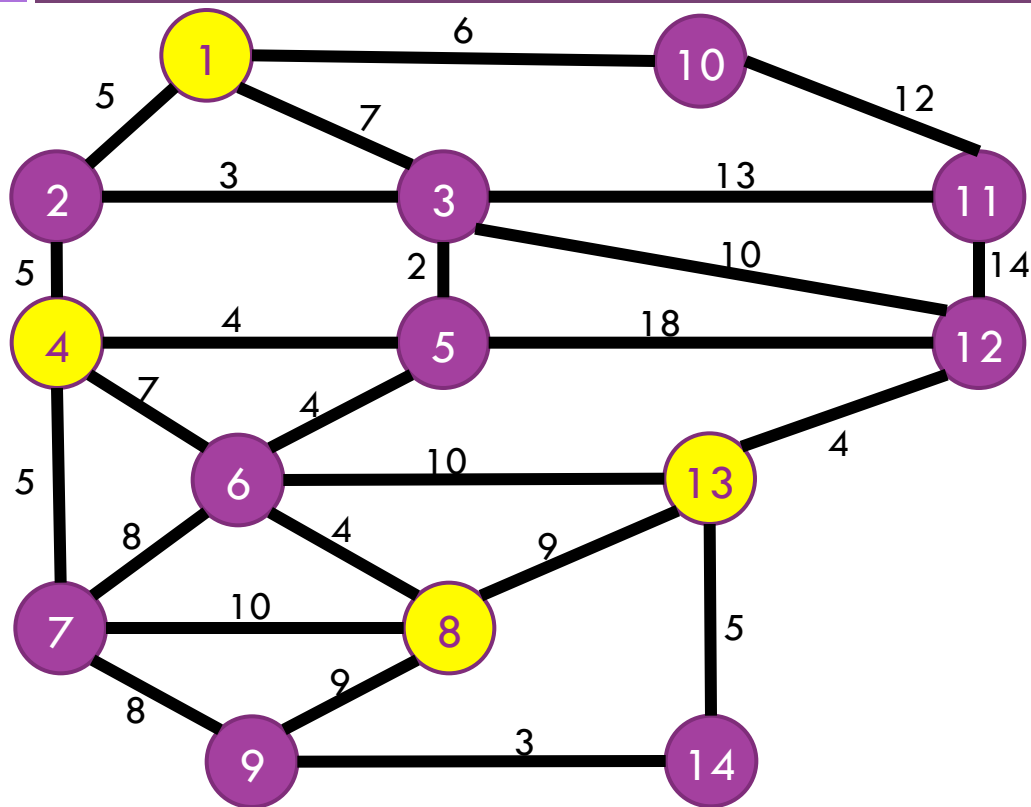
4) Substituir a AGM pelos caminhos reais





# Algoritmo de arborescências

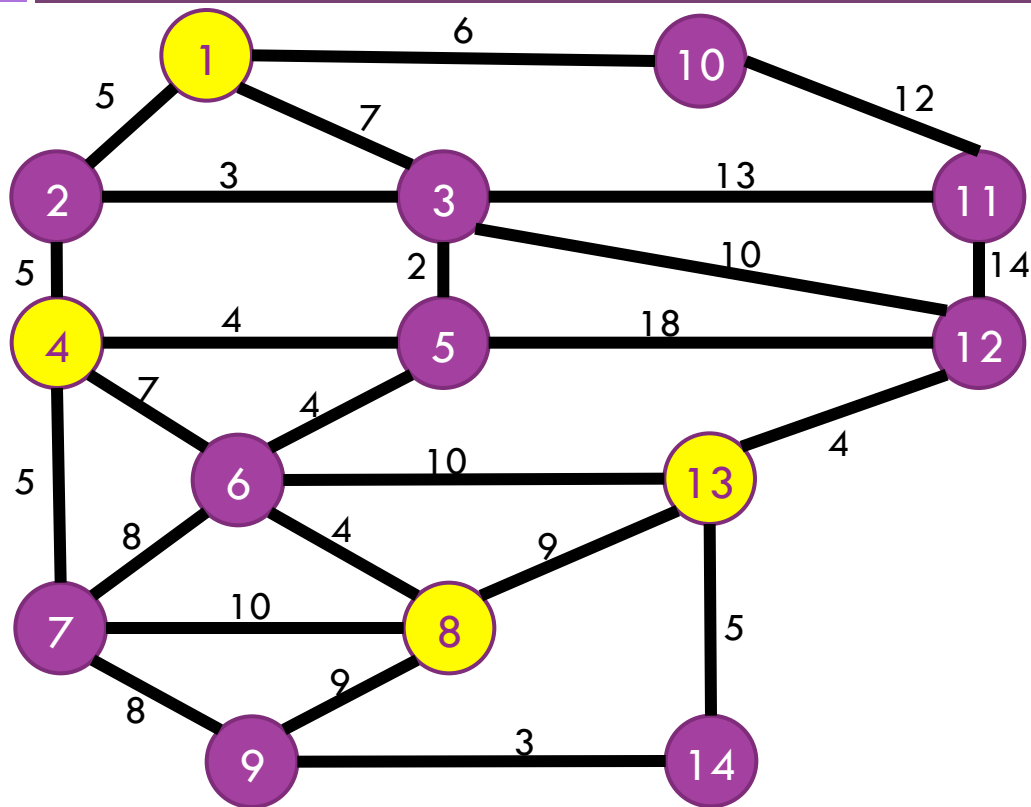
33



Arborescência: conectar árvores mínimas com raízes locais

# Algoritmo de arborescências

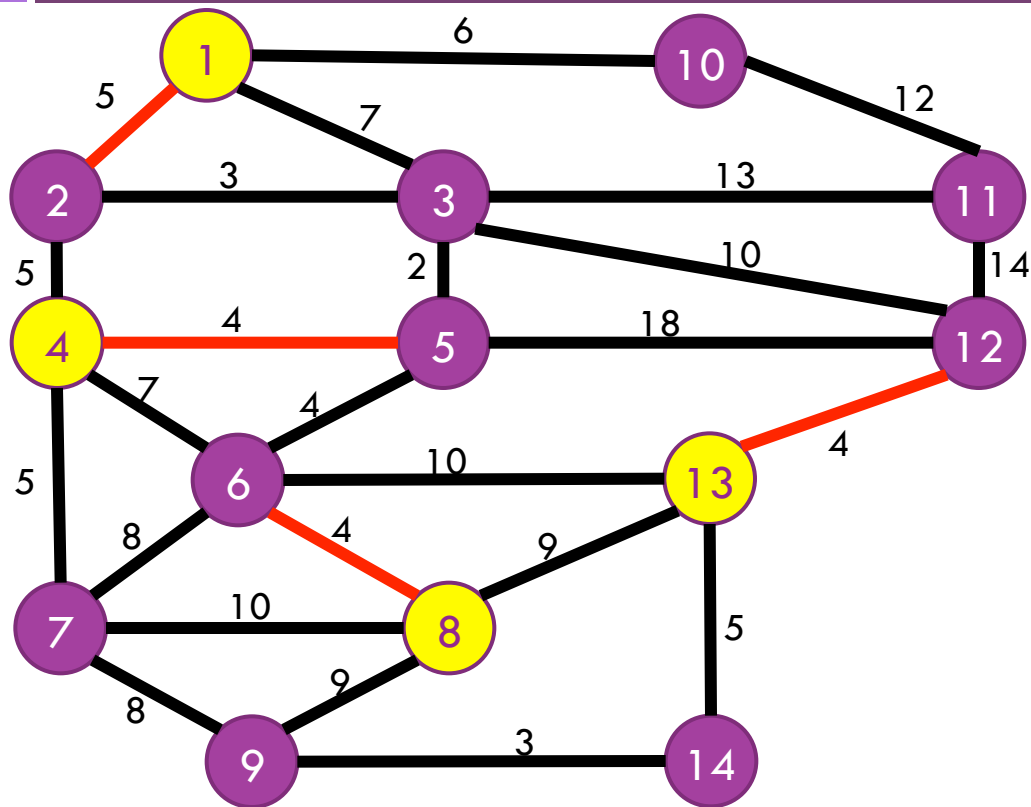
34



- 1) Determine o vértice mais próximo de cada terminal
- 2) Enquanto existir mais de uma arborescência:
  - Determinar o vértice mais próximo de cada arborescência
  - Unir arborescências com vértices em comum
- 3) Podar arestas desnecessárias

# Algoritmo de arborescências

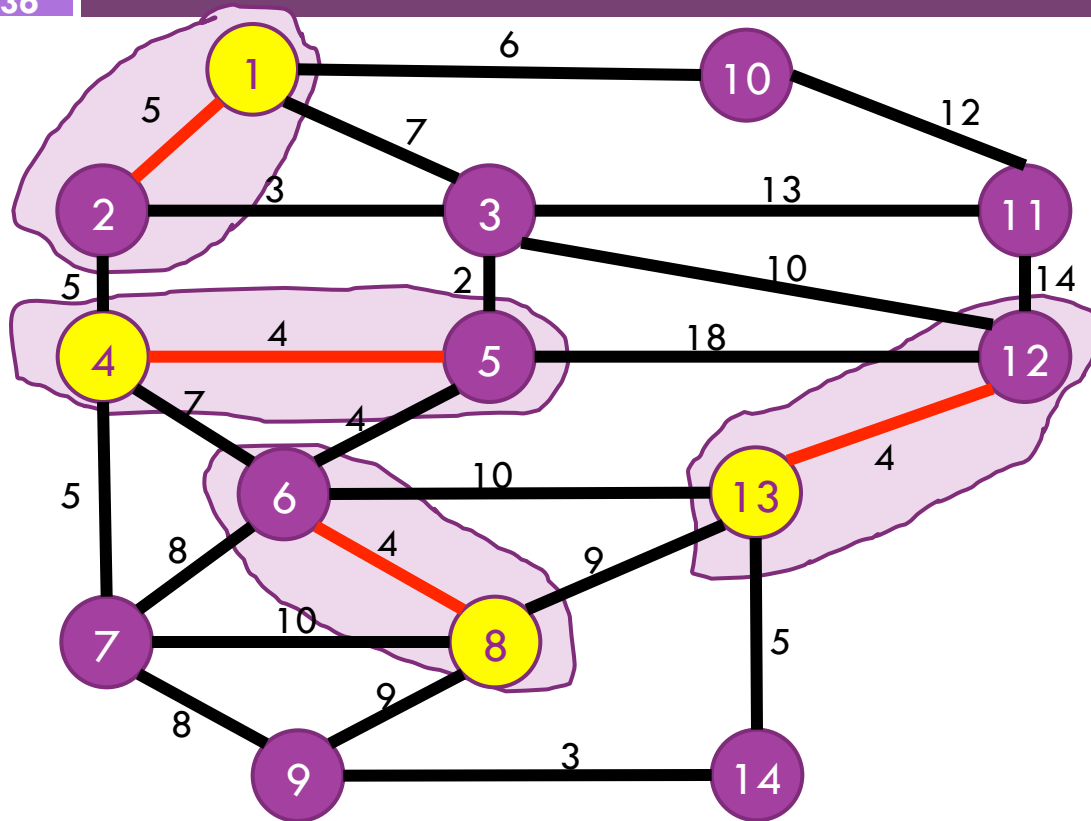
35



1) Determine o vértice mais próximo de cada terminal

# Algoritmo de arborescências

36



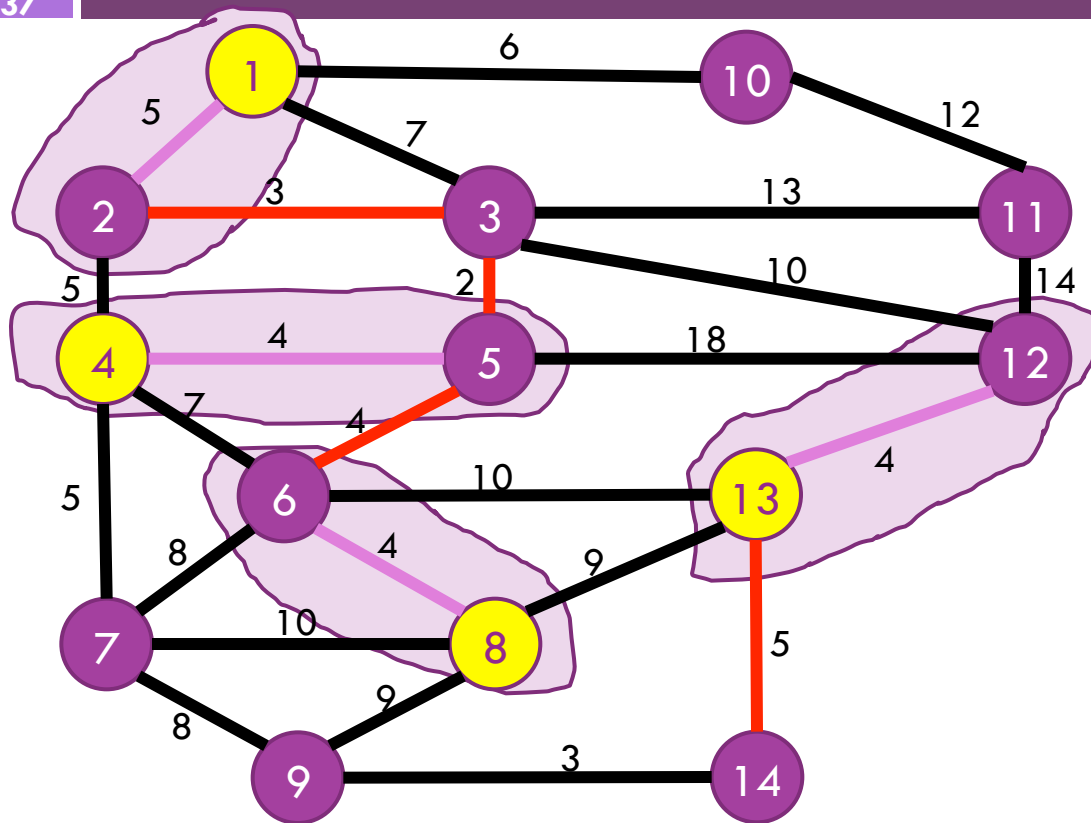
2) Enquanto existir mais de uma arborescência:

- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

37



2) Enquanto existir mais de uma arborescência:

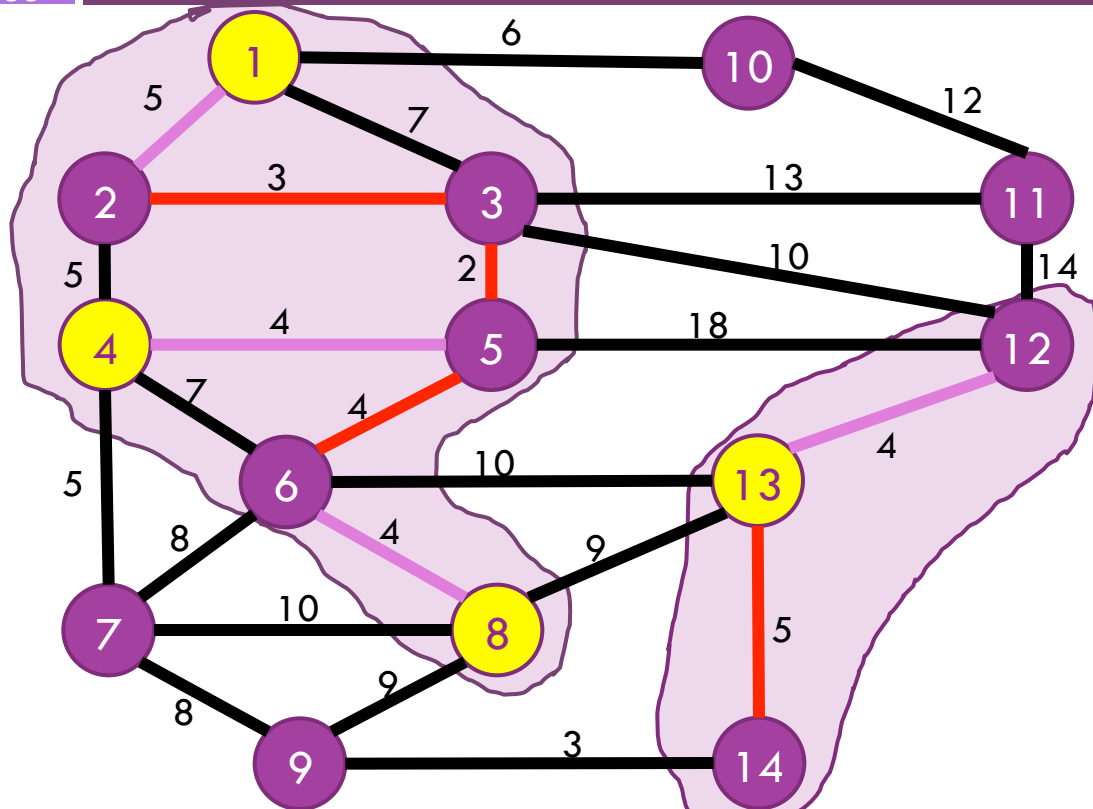
- Determinar o vértice mais próximo de cada arborescência

- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

38



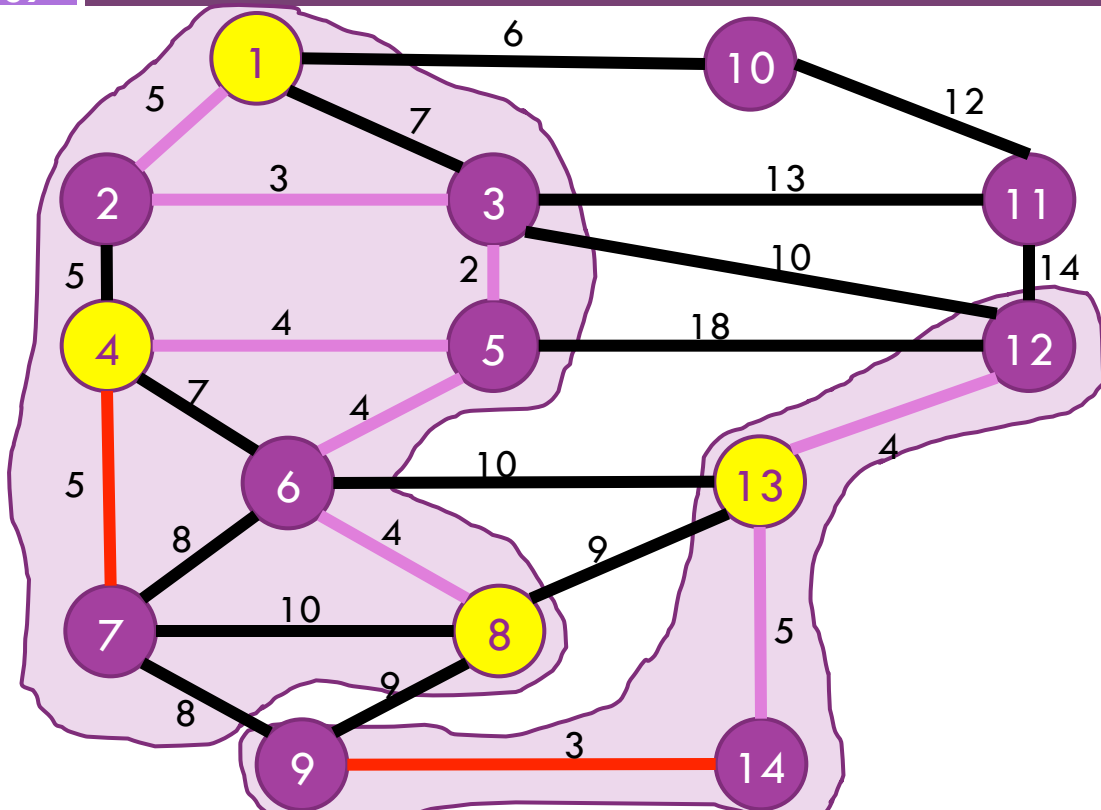
2) Enquanto existir mais de uma arborescência:

- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

39



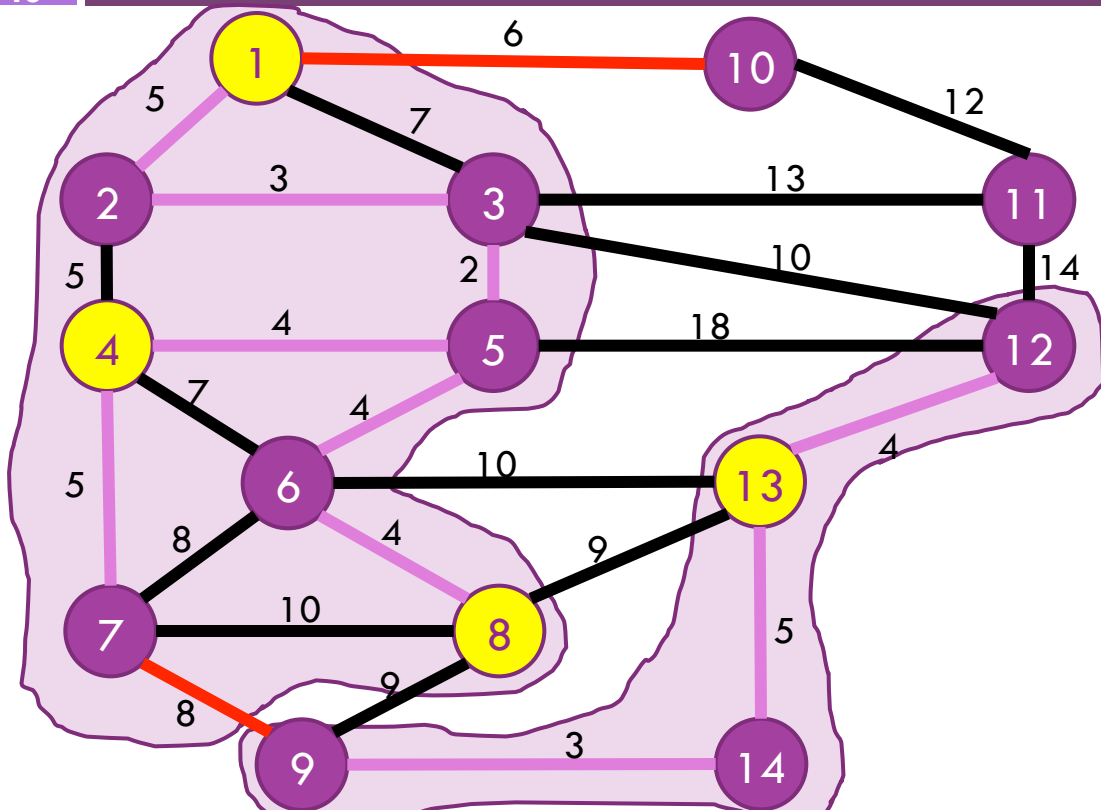
2) Enquanto existir mais de uma arborescência:

- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

40



2) Enquanto existir mais de uma arborescência:

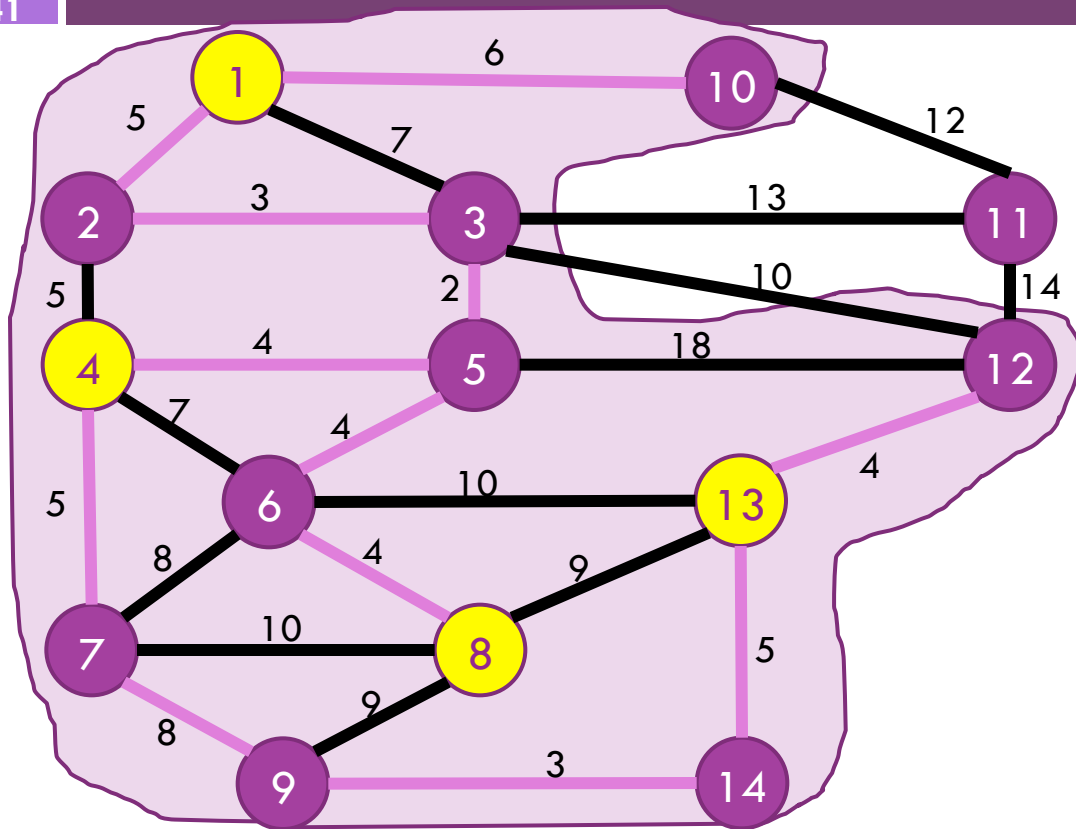
- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias



# Algoritmo de arborescências

41



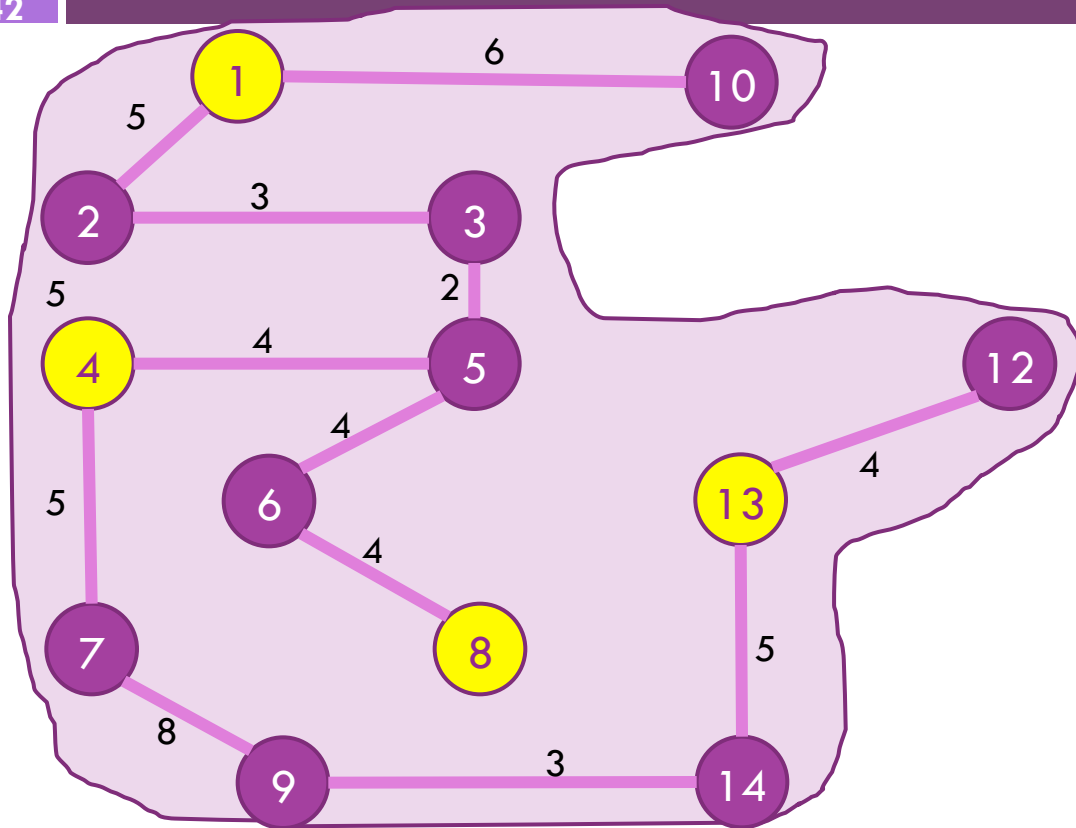
2) Enquanto existir mais de uma arborescência:

- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

42



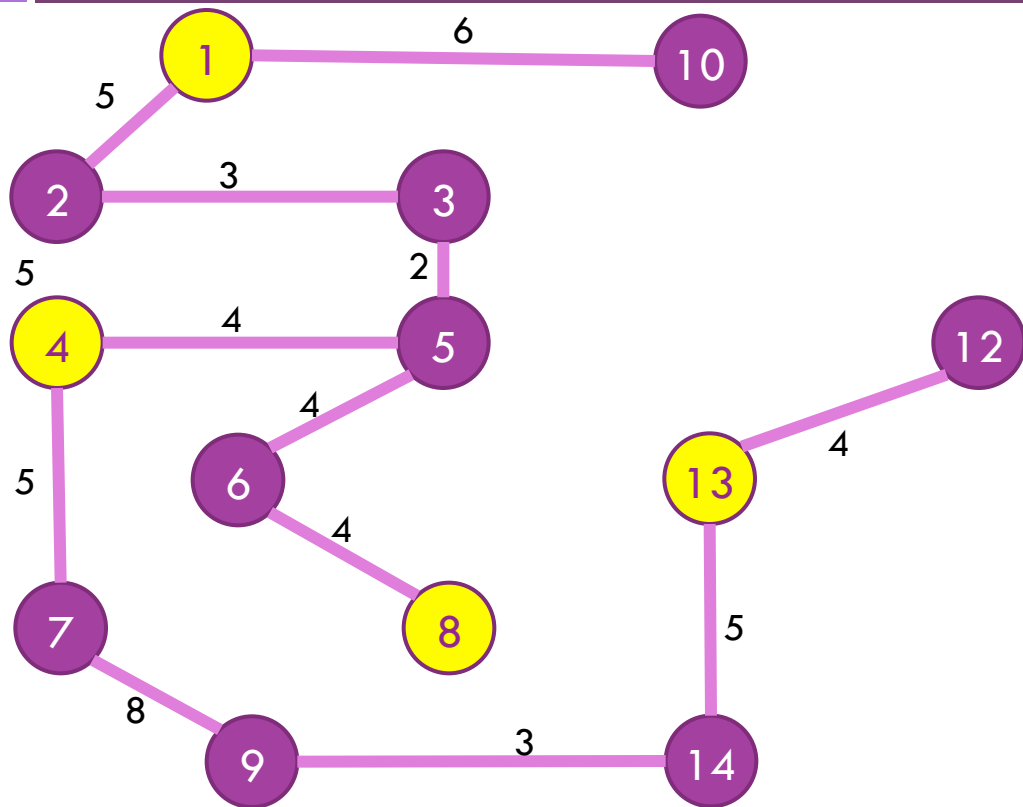
2) Enquanto existir mais de uma arborescência:

- Determinar o vértice mais próximo de cada arborescência
- Unir arborescências com vértices em comum

3) Podar arestas desnecessárias

# Algoritmo de arborescências

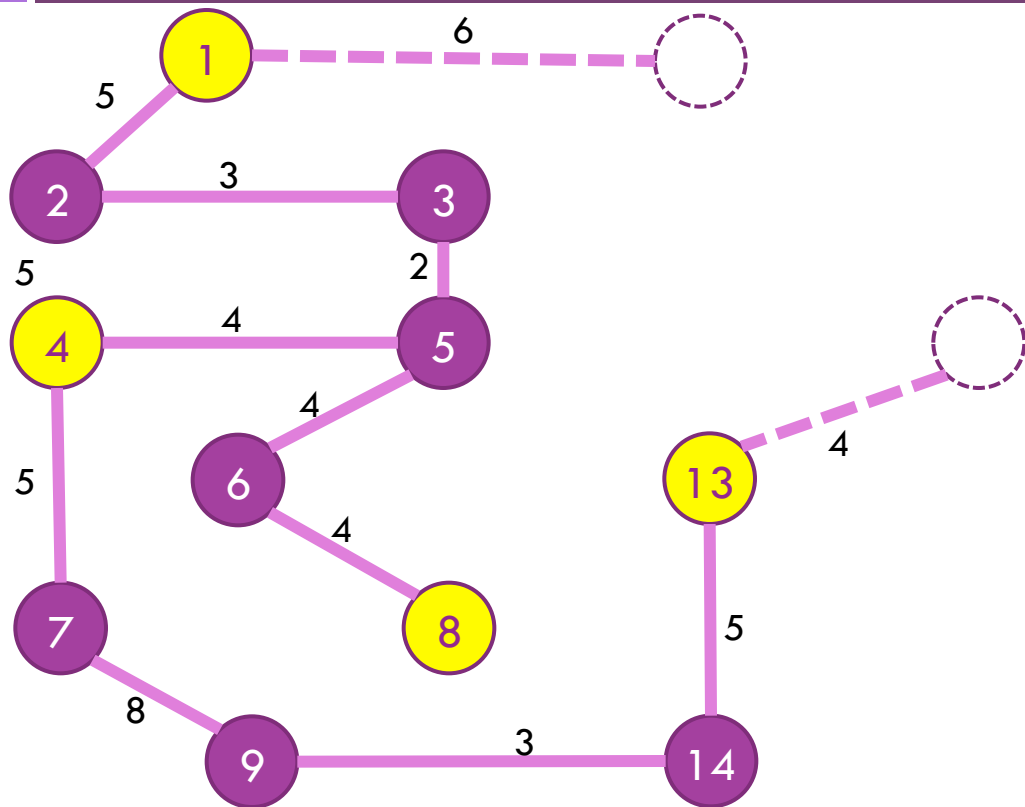
43



3) Podar arestas desnecessárias

# Algoritmo de arborescências

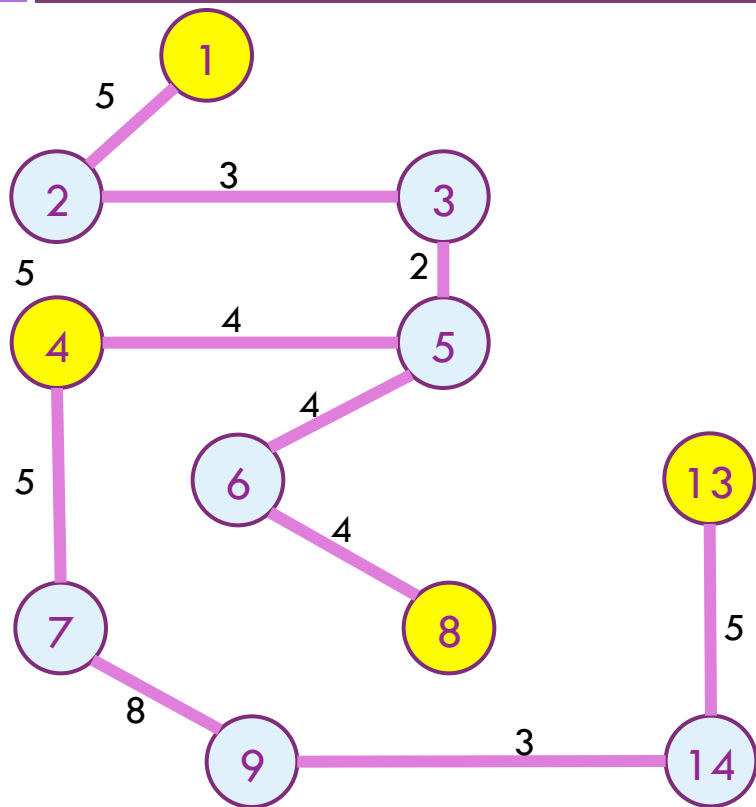
44



3) Podar arestas desnecessárias

# Algoritmo de arborescências

45

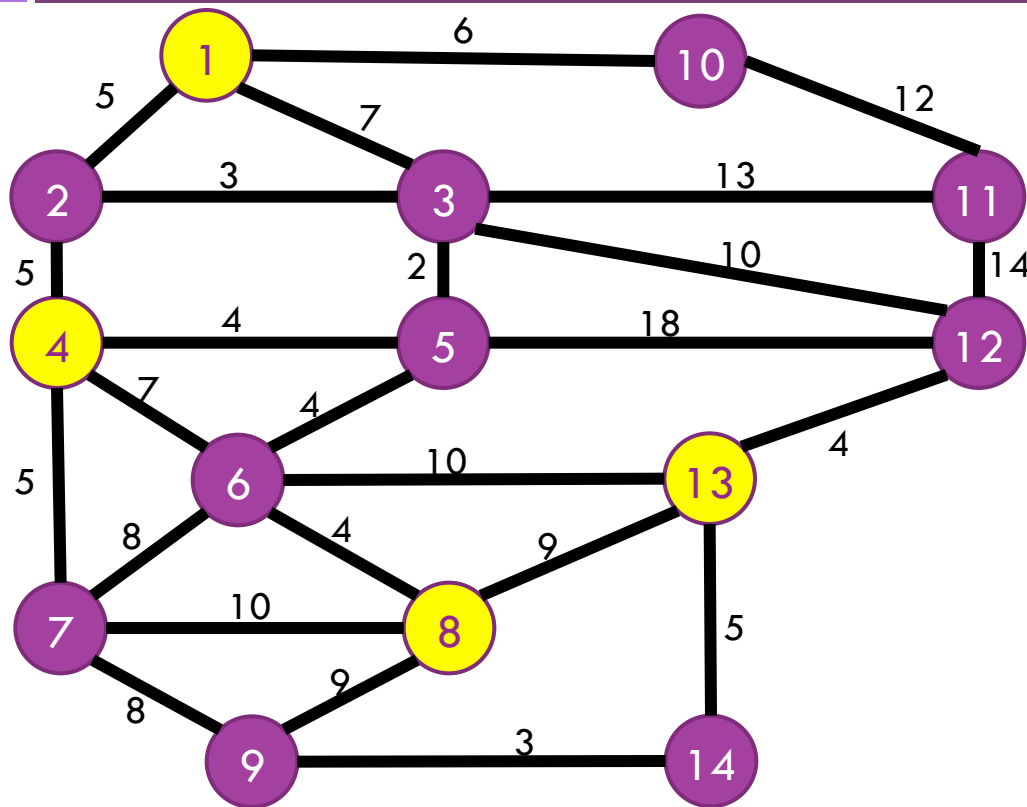


3) Podar arestas desnecessárias

Árvore de Steiner

# Algoritmo do vértice mais próximo

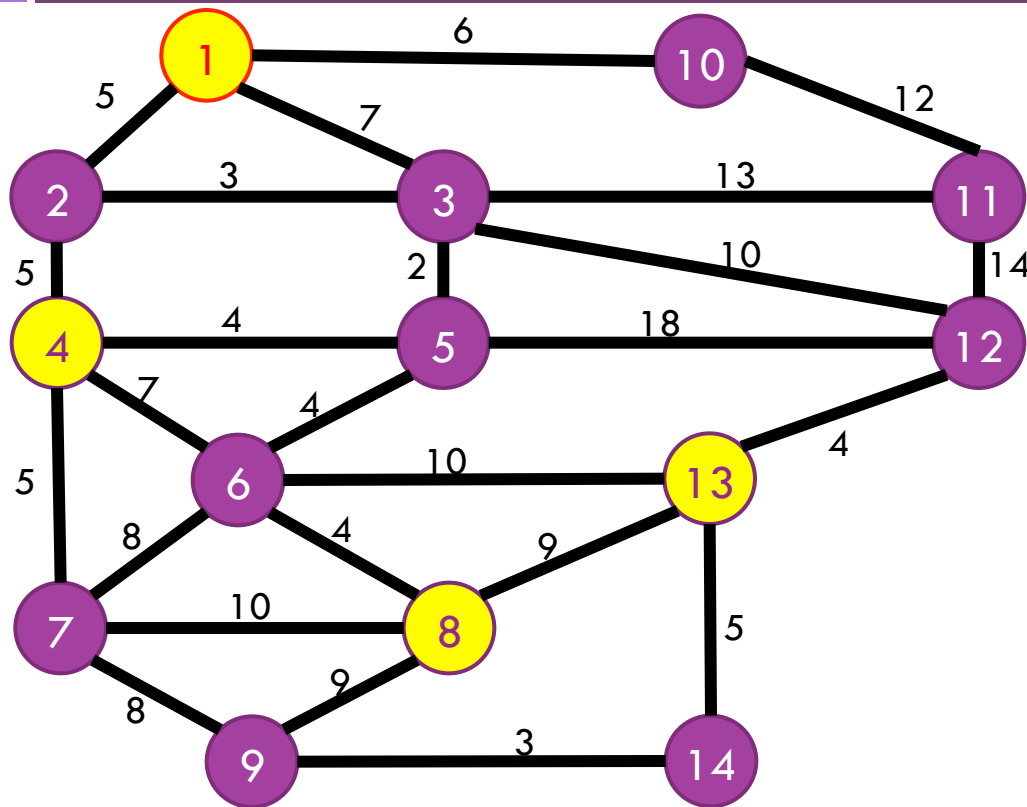
46



- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

# Algoritmo do vértice mais próximo

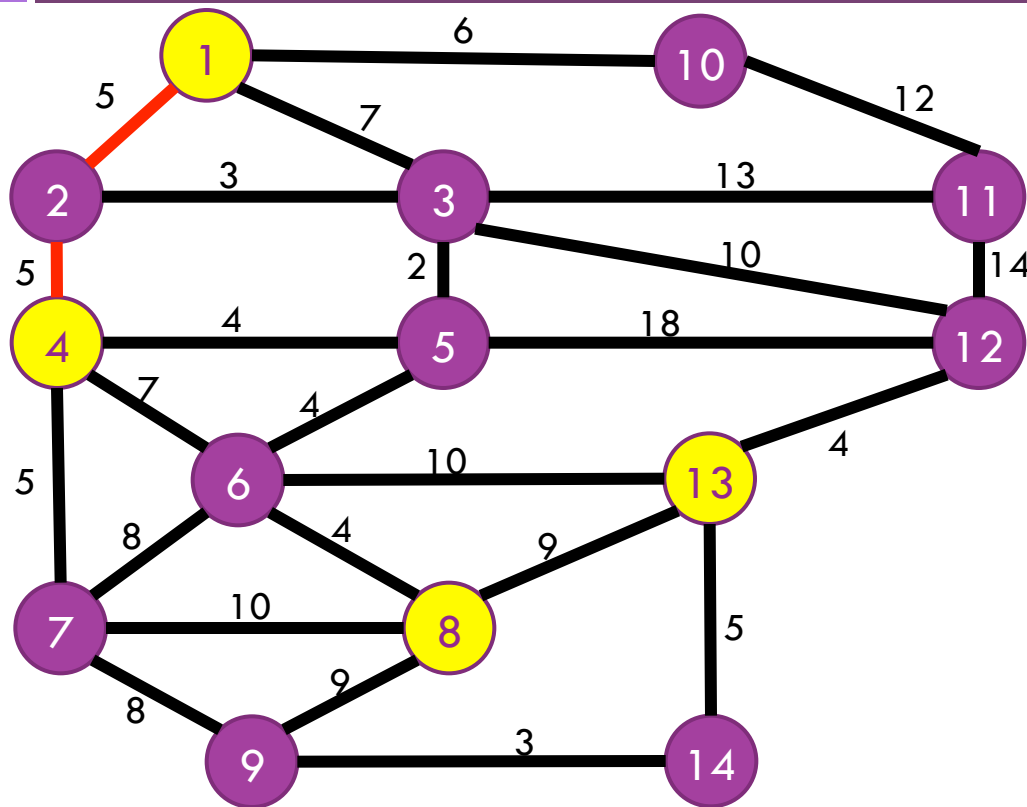
47



- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

# Algoritmo do vértice mais próximo

48

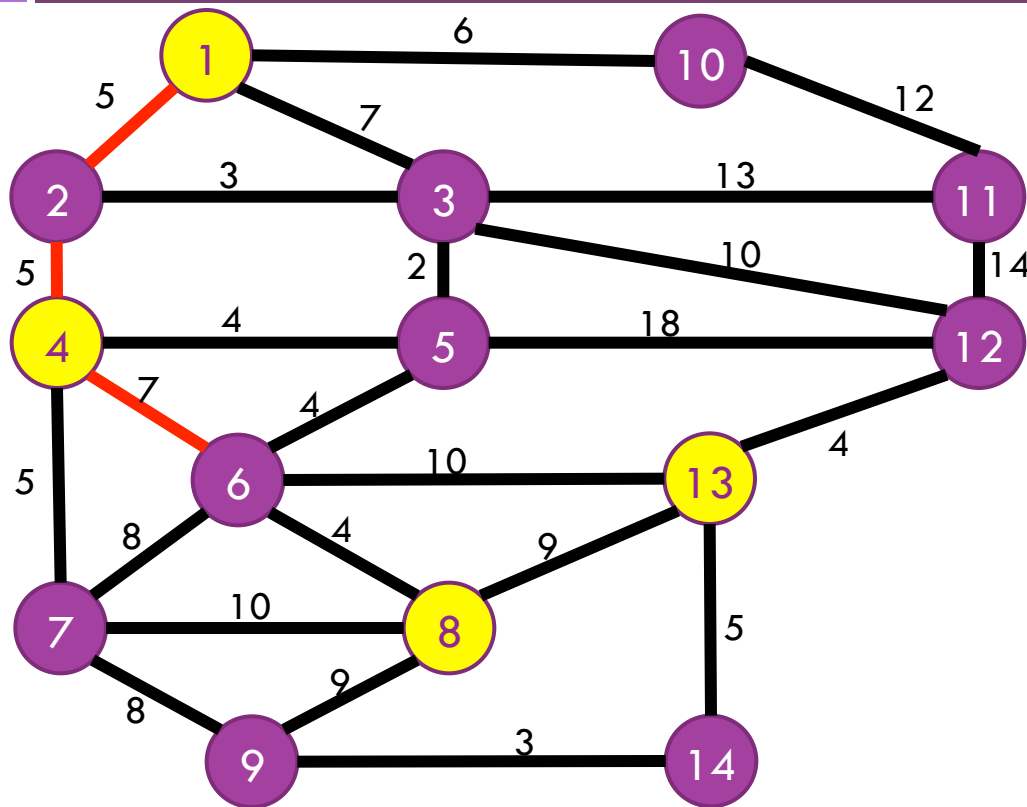


- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore



# Algoritmo do vértice mais próximo

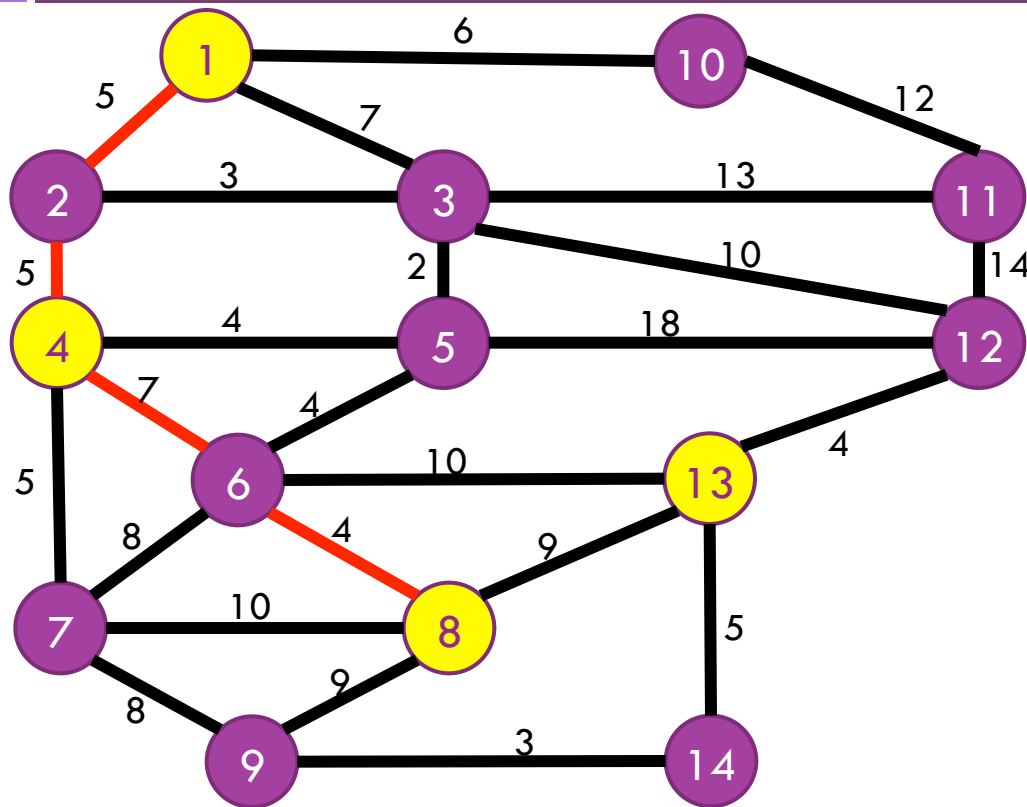
49



- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

# Algoritmo do vértice mais próximo

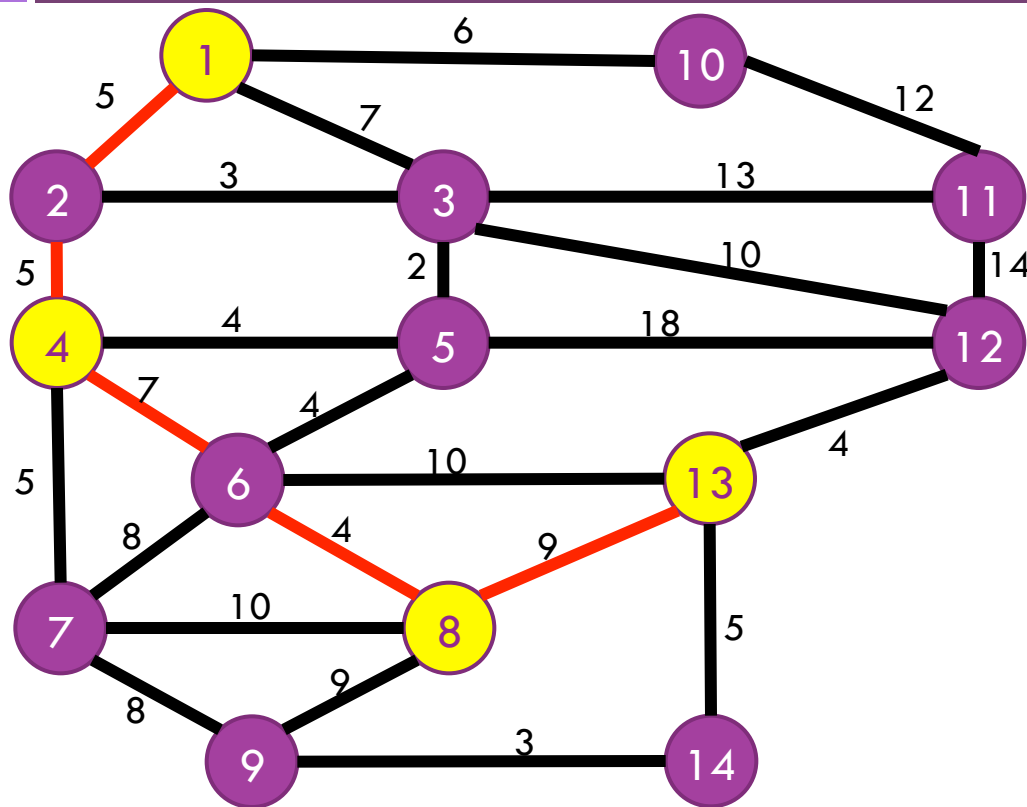
50



- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

# Algoritmo do vértice mais próximo

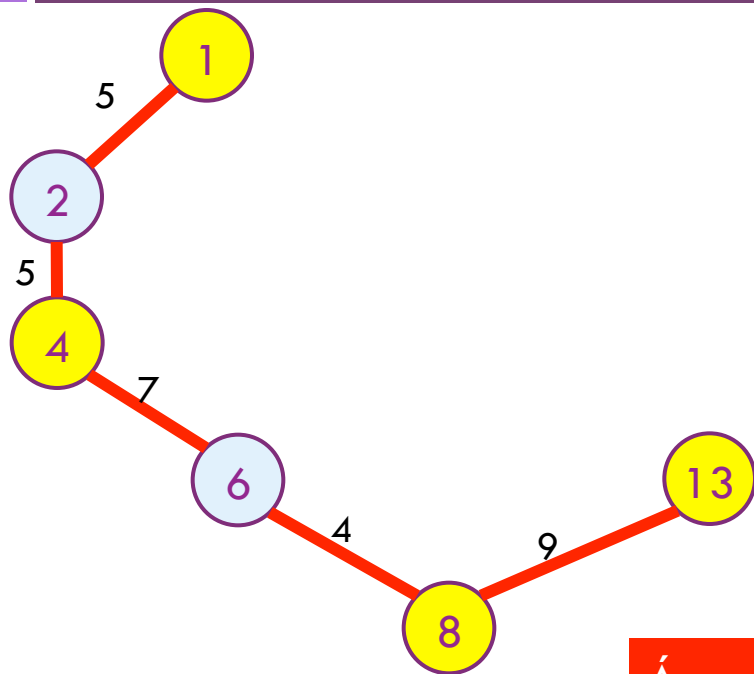
51



- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

# Algoritmo do vértice mais próximo

52



Árvore de Steiner

- 1) Construir a matriz de distância mínima entre todos os vértices (*Floyd-Warshall*)
- 2) Escolher um terminal
- 3) Ligar ao terminal mais próximo fora da árvore

OBRIGADO.

Dúvidas?