*Prof. Lucas de Souza Batista - DEE/EE/UFMG*

# Otimização de Redes

Fluxo Mínimo e
Fluxo Máximo

# Problema de Fluxo Mínimo

- O *problema de fluxo mínimo* consiste em determinar o valor do menor fluxo possível através de uma rede capaz de satisfazer uma demanda dada.

# Problema de Fluxo Mínimo

- Exemplos práticos:
  - mínimo fluxo de água através da rede de distribuição que atenda uma determinada demanda;
  - mínimo fluxo de gás/pretóleo através de um gasoduto que atenda uma demanda pré-estabelecida;
  - mínimo fluxo de carga em uma rede viária que atenda uma demanda definida;

# Problema de Fluxo Mínimo

- Elementos básicos:
  - $x_{ij}$: quantidade de fluxo no arco $(i,j)$;
  - $c_{ij}$: custo por unidade de fluxo no arco $(i,j)$;
  - $d_i$: demanda no nó $i$;
  - $u_{ij}$: limite máximo de fluxo no arco $(i,j)$;

# Formulação

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij}$$

$$\sum_{j\in S(i)} x_{ij} - \sum_{k\in P(i)} x_{ki} = d_i \qquad \forall i = 1, 2, \ldots, n$$

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in E$$

# Problema de Fluxo Máximo

- O *problema de fluxo máximo* consiste em determinar o valor do maior fluxo ($y$) possível que pode ser enviado de um nó a outro da rede.

# Problema de Fluxo Máximo

- Exemplos práticos:
  - maximizar o fluxo através da rede de distribuição de uma empresa, entre fábricas e clientes;
  - maximizar o fluxo através da rede de suprimentos de uma empresa, entre fornecedores e fábricas;
  - maximizar o fluxo de água/petróleo através de um sistema de tubulações;
  - maximizar o fluxo de veículos através de uma rede de transporte.

# Formulação

$$\max y$$

$$\sum_{j \in S(1)} x_{1j} - \sum_{k \in P(1)} x_{k1} = y$$

$$\sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = 0 \qquad i = 2, 3, \ldots, n-1$$

$$\sum_{j \in S(n)} x_{nj} - \sum_{k \in P(n)} x_{kn} = -y$$

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall (ij) \in E$$

# Algoritmo de Ford Fulkerson

Entradas:

- $G(N,E)$
- $u(i,j)$: capacidade máxima do arco $(i,j)$ $((u(i, j) > 0)$
- $1$: nó fonte
- $n$: nó sorvedouro

Saídas:

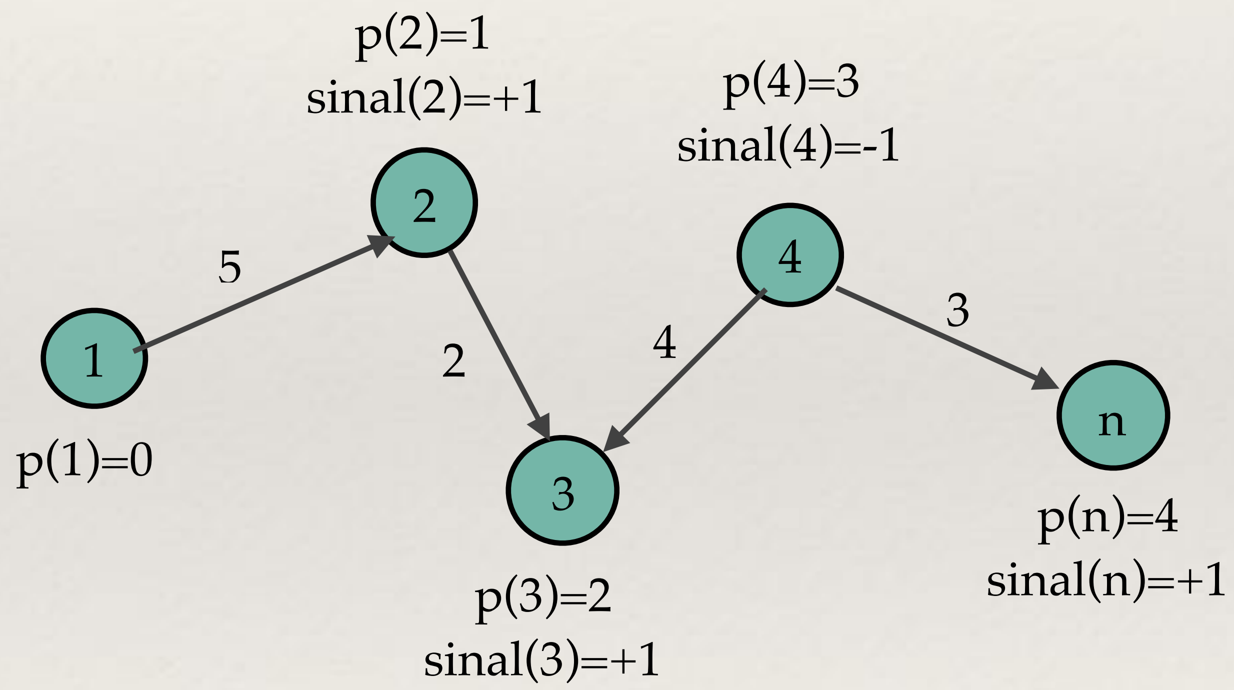
- Fluxo máximo $y$ do nó $1$ ao nó $n$

1. $y \leftarrow 0$. *(fluxo inicial)*
2. $v^{+}(i,j) \leftarrow u(i,j)\ \forall (i,j) \in E$. *(cap. máx. de aumento de fluxo)*
3. $v^{-}(i,j) \leftarrow 0\ \forall (i,j) \in E$. *(cap. máx. de diminuição de fluxo)*
4. $R \leftarrow \{n\}$. *(conjunto de nós rotulados)*
5. <u>**Enquanto**</u> $n \in R$:
   A. <u>**Execute**</u> <u>**CONSTROI_CAMINHO**</u>.
   B. <u>**Se**</u> $n \in R$:
      I. <u>**Execute**</u> <u>**AUMENTA_FLUXO**</u>.

# CONSTROI_CAMINHO

**Entradas:**

- $G(N,E)$
- $v^{+}(i,j)$
- $v^{-}(i,j)$
- $1$: nó fonte
- $n$: nó sorvedouro
- $R$




**Saídas:**

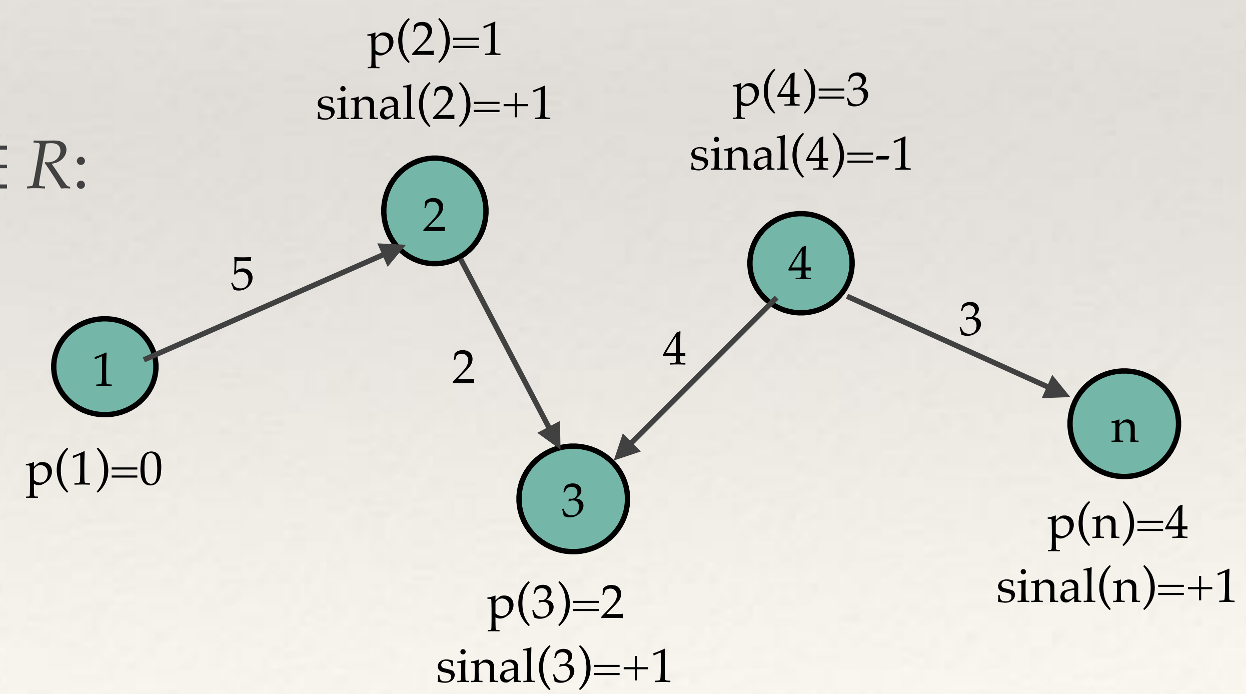
- $R$
- $p(i)$
- *sinal(i) (igual a +1 se arco (p(i),i), ou igual a -1 se arco (i,p(i)))*

## Passo 1: Início

1. $p(i) \leftarrow 0 \ \forall i \in N$. *(predecessor)*
2. $R \leftarrow \{1\}$.
3. $Lista \leftarrow \{1\}$.

## Passo 2:

1. **Enquanto** $(Lista \neq \emptyset) \vee (n \notin R)$:
    A. Escolha qualquer $i \in Lista$.
    B. $Lista \leftarrow Lista - \{i\}$.
    C. **Para todo** $(i,j) \in E \mid v^{+}(i,j) > 0 \wedge j \notin R$:
        I. $p(j) \leftarrow i$.
        II. $sinal(j) \leftarrow +1$.
        III. $R \leftarrow R + \{j\}$.
        IV. $Lista \leftarrow Lista + \{j\}$.
    D. **Para todo** $(j,i) \in E \mid v^{-}(j,i) > 0 \wedge j \notin R$:
        I. $p(j) = i$.
        II. $sinal(j) = -1$.
        III. $R = R + \{j\}$.
        IV. $Lista = Lista + \{j\}$.



# AUMENTA_FLUXO

**Entradas:**

- *G(N,E)*
- $v^{+}(i,j)$
- $v^{-}(i,j)$
- *1*: nó fonte
- *n*: nó sorvedouro
- *y*: fluxo atual
- *p(i)*: predecessor
- *sinal(i)*: *(se +1: (p(i),i), ou se -1: (i,p(i)))*

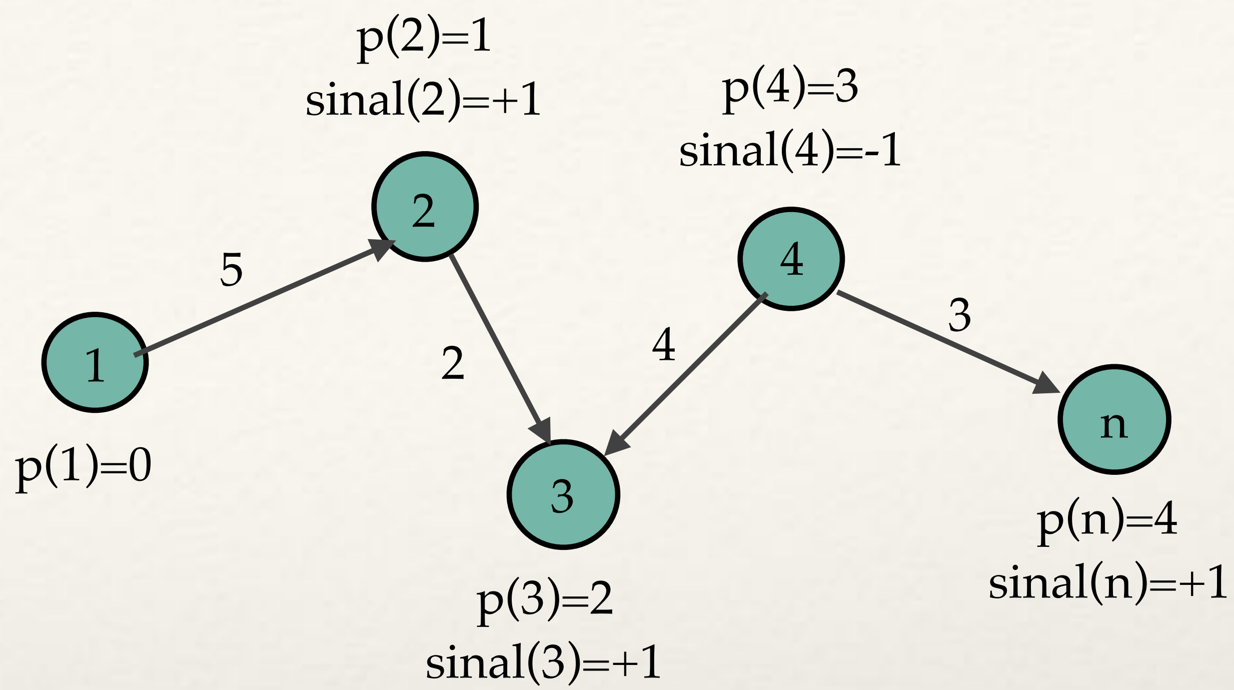
**Saídas:**

- *y*: fluxo aumentado
- $v^{+}(i,j)$
- $v^{-}(i,j)$

## Passo 1: Início

1. $r \leftarrow n$.
2. $C \leftarrow \emptyset$.
3. $\Delta \leftarrow \infty$.

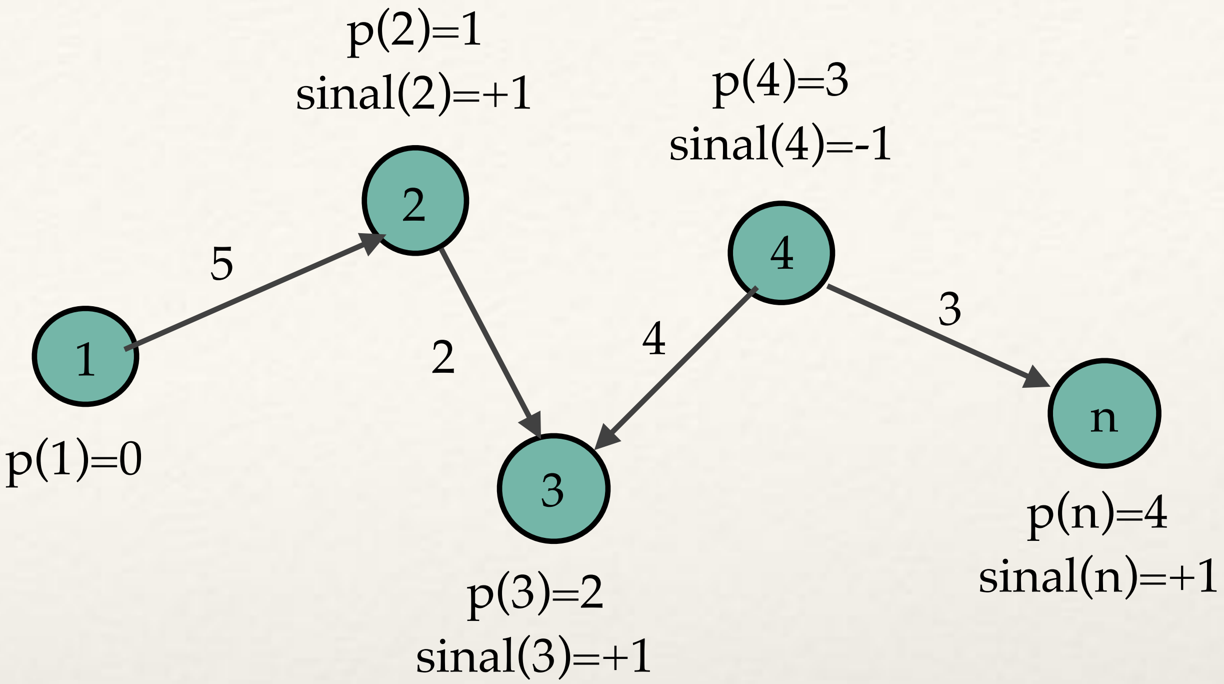
## Passo 2: Identificando $\Delta$



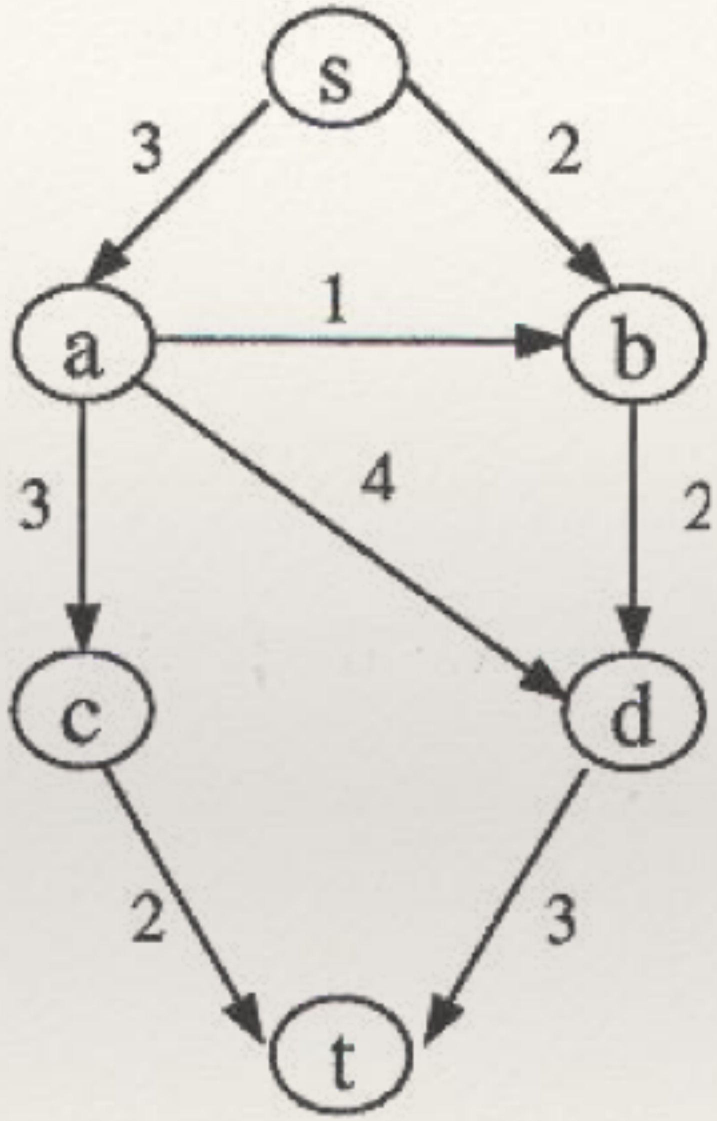

1. **<u>Enquanto</u>** r $\neq$ *1*:
   A. **<u>Se</u>** *sinal(r)* = +*1*:
      I. $C \leftarrow C + \{(p(r),r)\}$.
      II. $\Delta \leftarrow \min(\Delta, v^+(p(r),r))$.
   B. **<u>Se não</u>**:
      I. $C \leftarrow C + \{(r,p(r))\}$.
      II. $\Delta \leftarrow \min(\Delta, v^-(r,p(r)))$.
2. $r \leftarrow p(r)$.

## Passo 3: Atualizando o grafo

1. $y \leftarrow y + \Delta$.
2. **Para todo** $(i,j) \in C$:
    - A. **Se** $sinal(j) = +1$:
        - I. $v^+(i,j) \leftarrow v^+(i,j) - \Delta$.
        - II. $v^-(i,j) \leftarrow v^-(i,j) + \Delta$.
    - B. **Se não:**
        - I. $v^-(i,j) \leftarrow v^-(i,j) - \Delta$.
        - II. $v^+(i,j) \leftarrow v^+(i,j) + \Delta$.

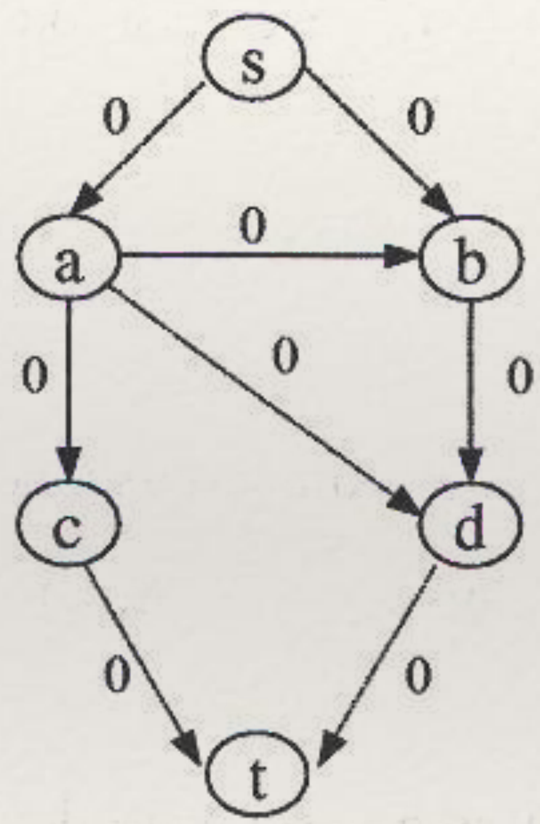


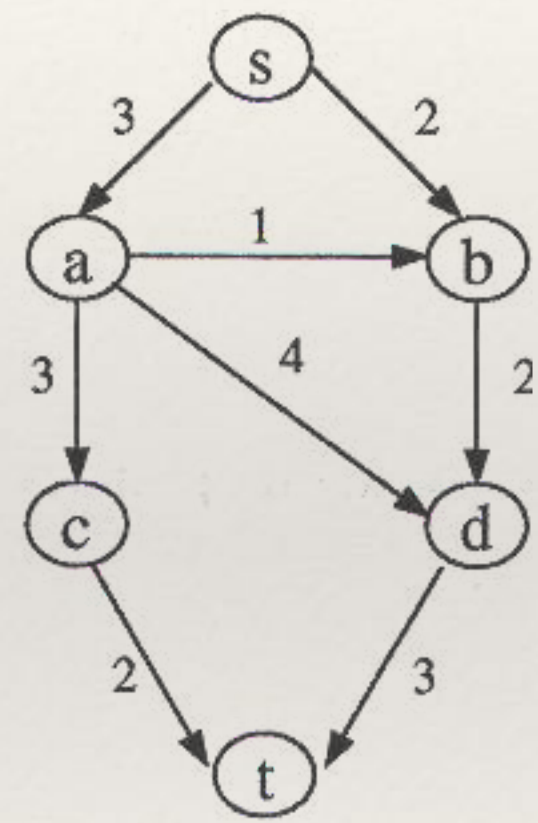
# Exemplo



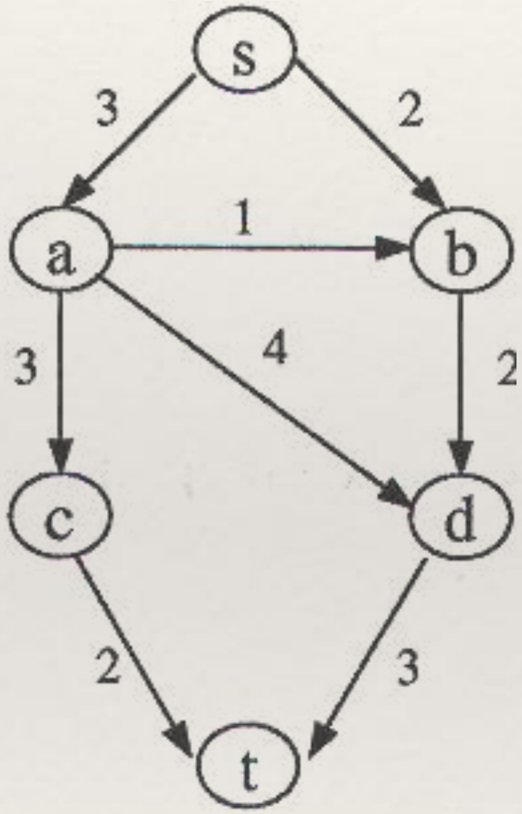

G

# Estado Inicial



# Iteração 1



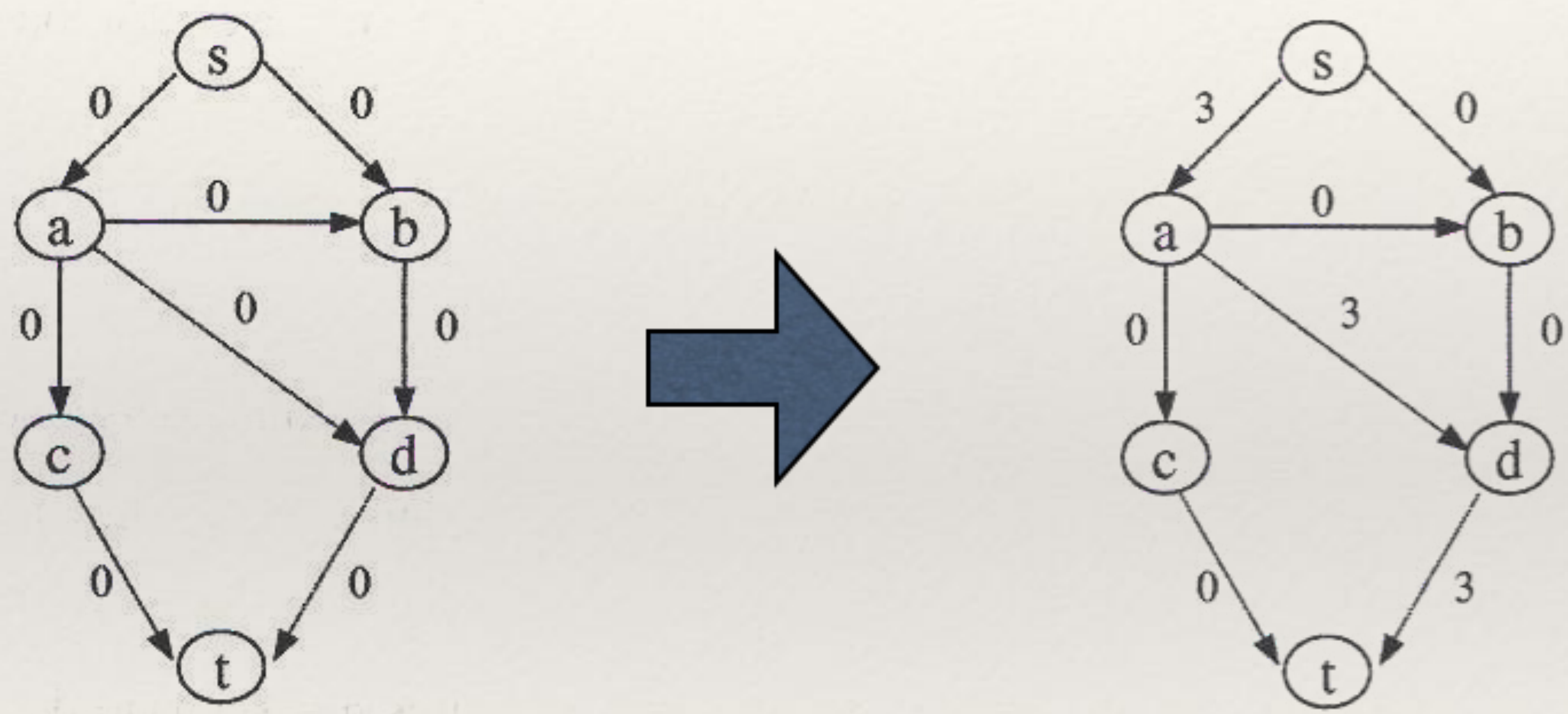

**Caminho:**

- *s-a-d-t*

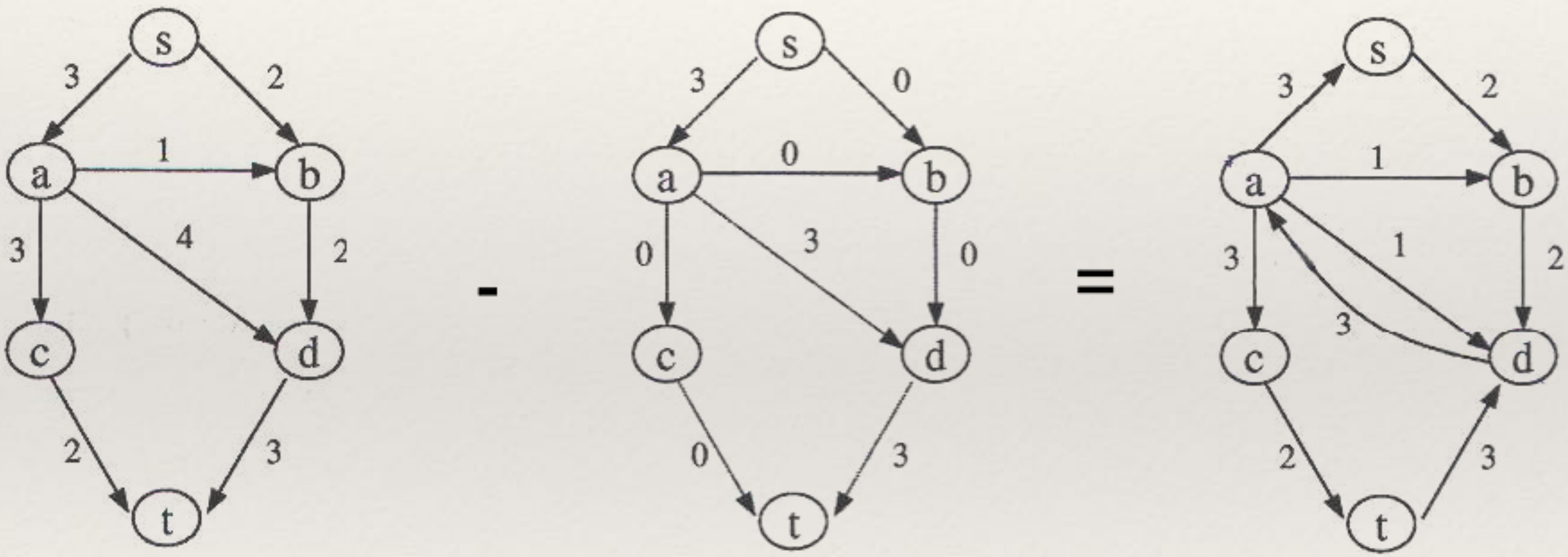
**Aumento máximo de fluxo:**

- 3

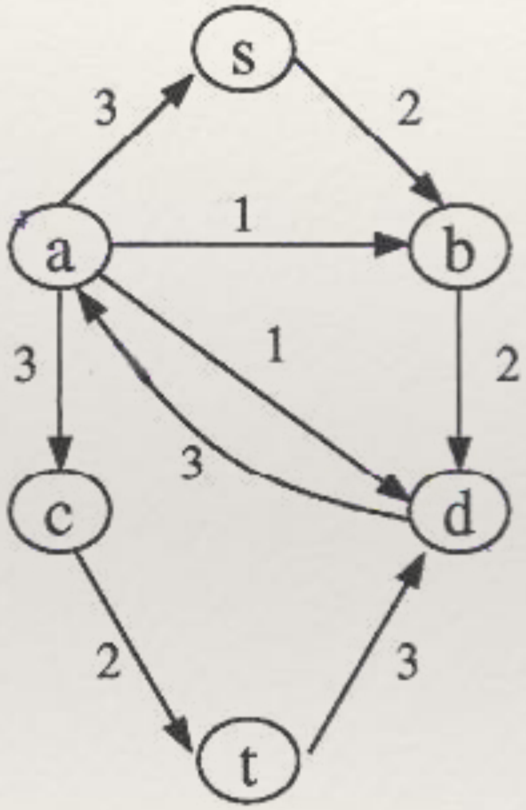
# Iteração 1



# Iteração 1



# Iteração 2



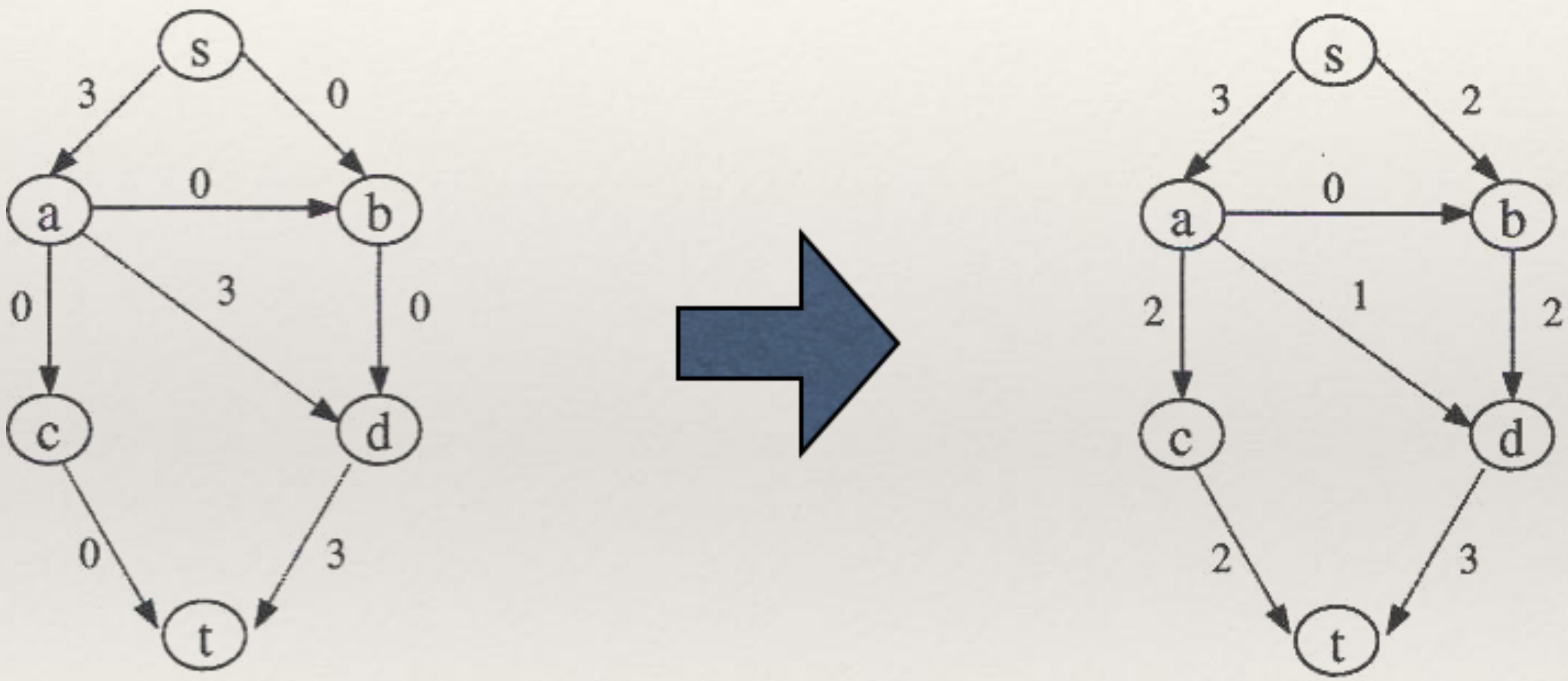

**Caminho:**

- *s-b-d-a-c-t*

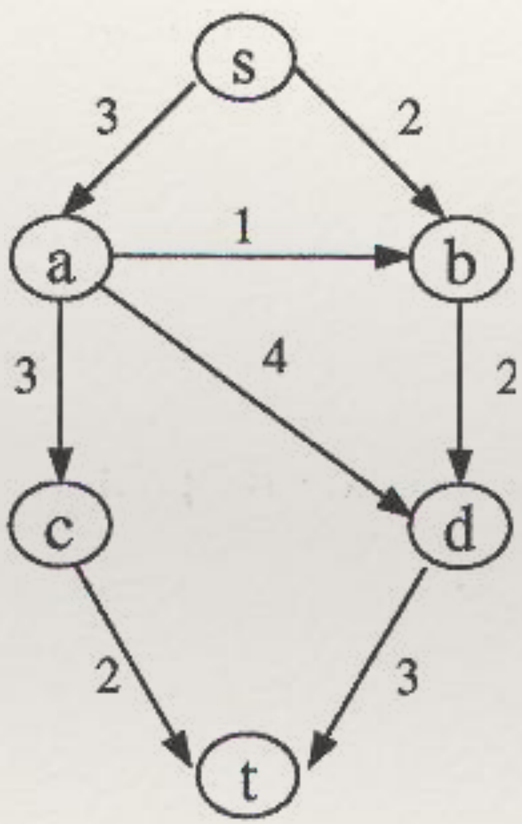
**Aumento máximo de fluxo:**

- 2

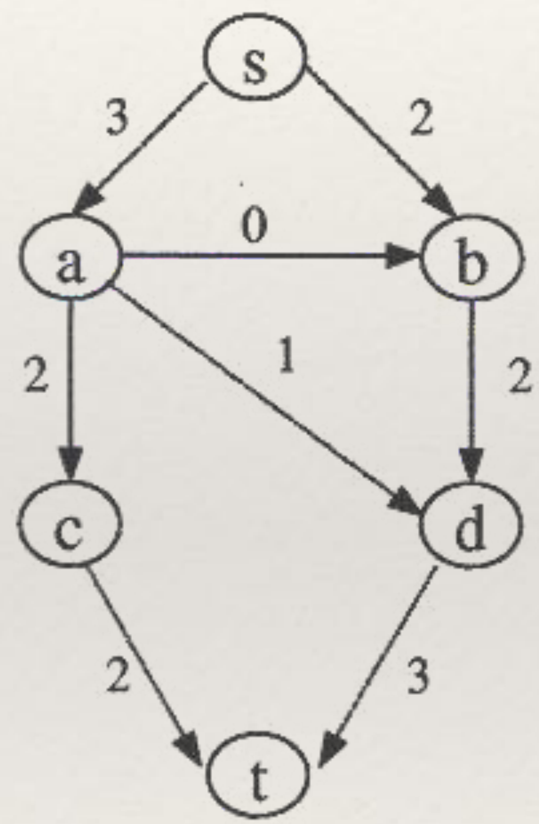
# Iteração 2



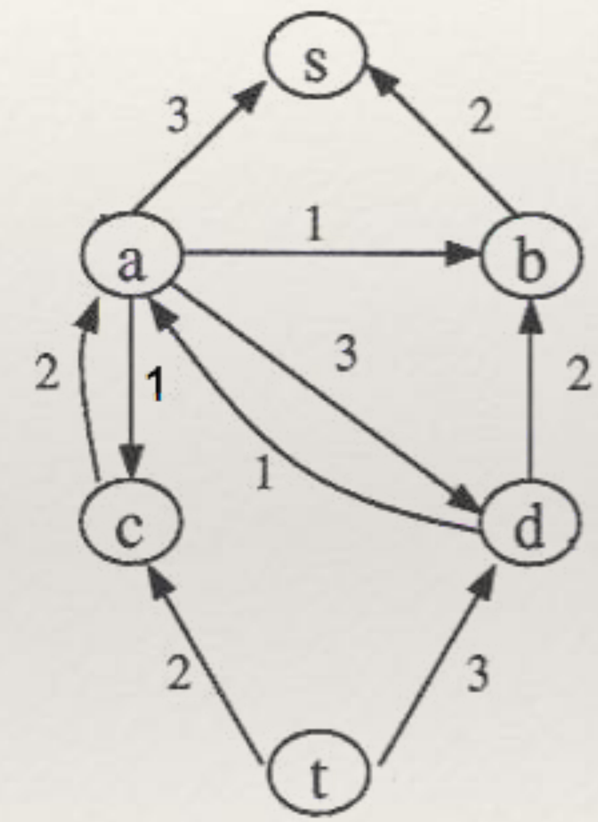
# Iteração 2



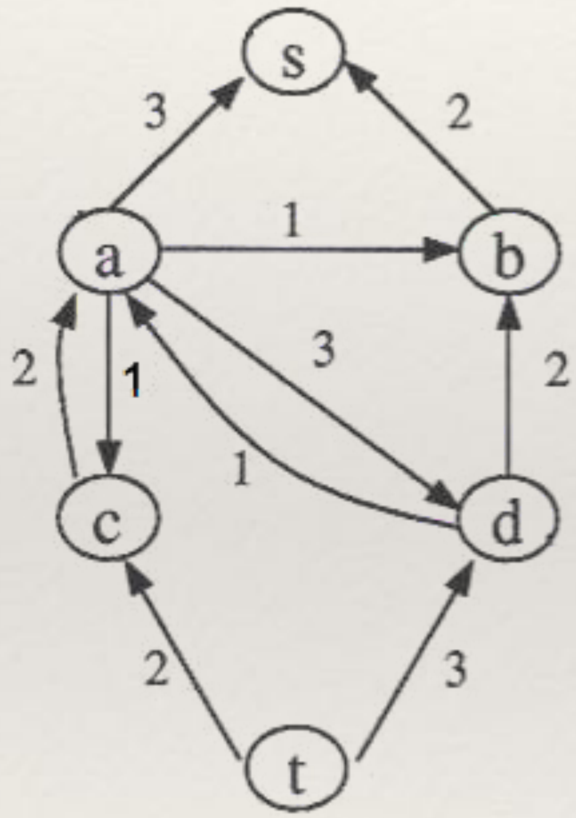

-




=

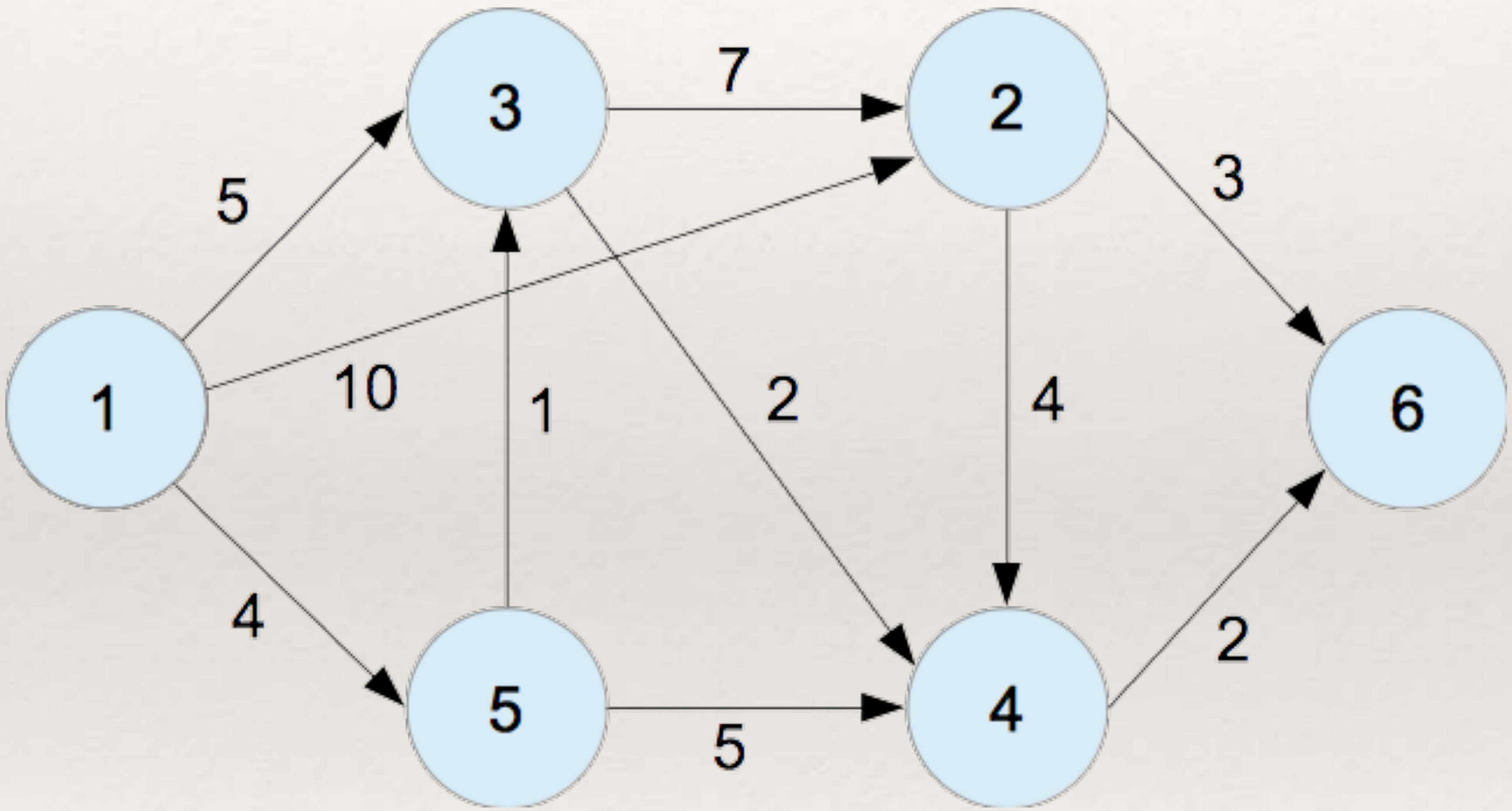


# Iteração 3




**Não existe caminho entre *s* e *t*.**

# Exercício



# Referências


- [Arenales et al, 2007] M. Arenales; V. Armentano; R. Morabito; H. Yanasse. **Pesquisa Operacional para Cursos de Engenharia**, Editora Campus / Elsevier, 2007.
- [Goldbarg et al, 2005] M. C. Goldbarg; H. P. Luna. **Otimização Combinatória e Programação Linear - Modelos e Algoritmos**, 2a ed., Editora Campus / Elsevier, 2005.
- [Hillier et al, 2013] F. S. Hillier; G. J. Lieberman. **Introdução à Pesquisa Operacional**, 9a ed., Editora Mc Graw Hill, 2013.