GRAFOS

ESTRUTURAS DE DADOS
BÁSICAS

Aula 1.3

Matriz de Adjacência

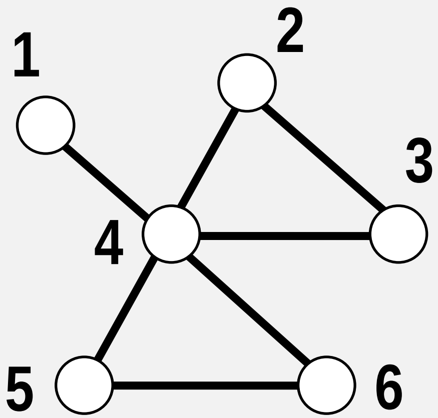
Uma matriz A=[a_{ij}] quadrada de ordem *n* é denominada Matriz de Adjacência de $G = (N,M)$ quando:

a_{ij}=1, se $(i,j) \in M$

a_{ij}=0 em caso contrário.

Exemplo: Grafo não direcionado

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						



Matriz de Adjacência

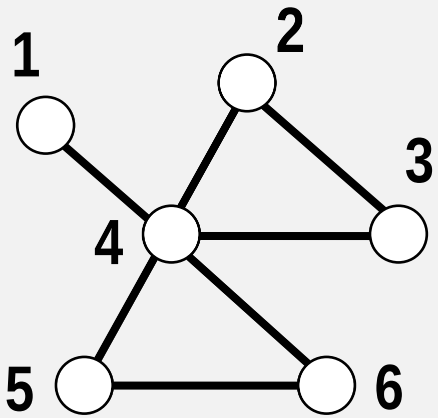
Uma matriz A=[a_{ij}] quadrada de ordem *n* é denominada Matriz de Adjacência de *G = (N,M)* quando:

$$a_{ij}=1, \text{ se } (i,j)\in M$$

$$a_{ij}=0 \text{ em caso contrário.}$$

Exemplo: Grafo não direcionado

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$



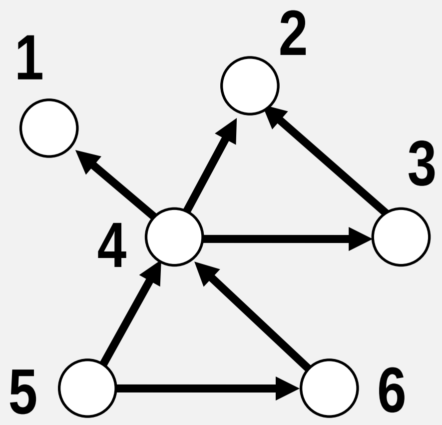
Matriz de Adjacência

Exemplo: Grafo direcionado

$a_{ij}=1$, se $(i,j) \in M$

$a_{ij}=0$ em caso contrário.

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						



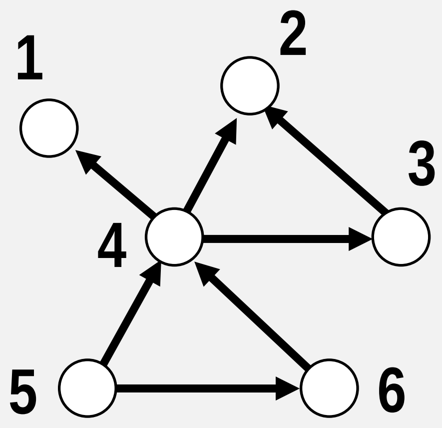
Matriz de Adjacência

Exemplo: Grafo direcionado

$$a_{ij}=1,\ \text{se}\ (i,j) \in M$$

$$a_{ij}=0\ \text{em caso contrário.}$$

$$\begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$



Matriz de Incidência

Uma matriz $A=[a_{ki}]$ de dimensão *m x n* é denominada Matriz de Incidência de um grafo *G=(N,M)* quando:

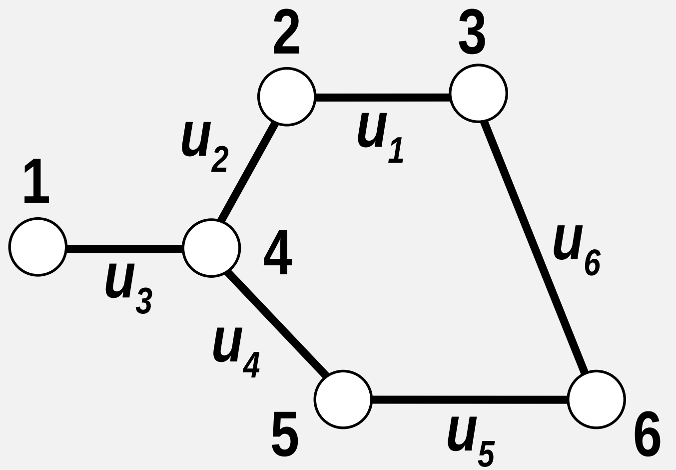
a_{ki} =+1, se a aresta u_k tem origem no vértice *i*

a_{ki} =-1, se *i* é o vértice destino da aresta u_k

a_{ki} = 0, se a aresta u_k não incide no vértice *i*

Exemplo: Grafo não direcionado

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{matrix} & \begin{bmatrix} \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; \\ \; & \; & \; & \; & \; & \; \end{bmatrix} \end{array}$$



Matriz de Incidência

Uma matriz $A=[a_{ki}]$ de dimensão $m \times n$ é denominada Matriz de Incidência de um grafo $G=(N,M)$ quando:

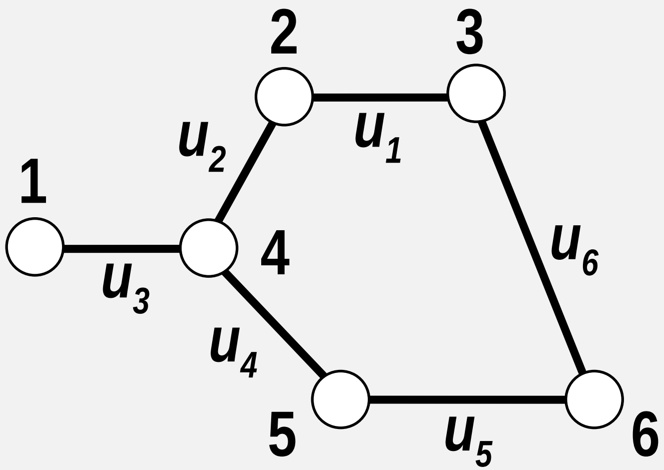
a_{ki} =+1, se a aresta u_k tem origem no vértice i

a_{ki} =-1, se i é o vértice destino da aresta u_k

a_{ki} = 0, se a aresta u_k não incide no vértice i

Exemplo: Grafo não direcionado

$$
\begin{array}{c}
\\
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$



Matriz de Incidência

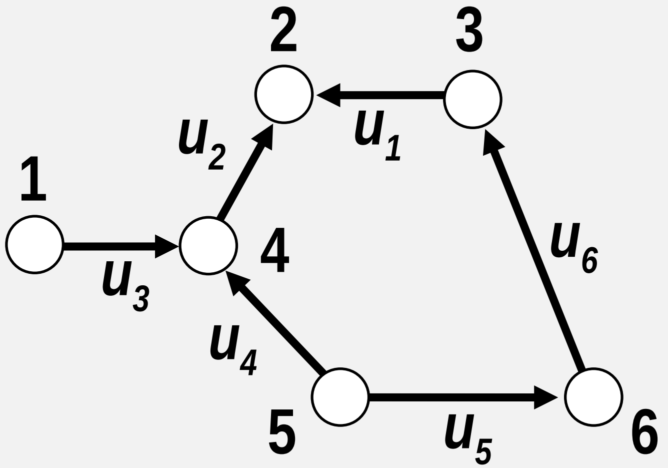
Exemplo: Grafo direcionado

a_{ki} =+1, se a aresta u_k tem origem no vértice i

a_{ki} =-1, se i é o vértice destino da aresta u_k

a_{ki} = 0, se a aresta u_k não incide no vértice i

	1	2	3	4	5	6
u_1						
u_2						
u_3						
u_4						
u_5						
u_6						



Matriz de Incidência

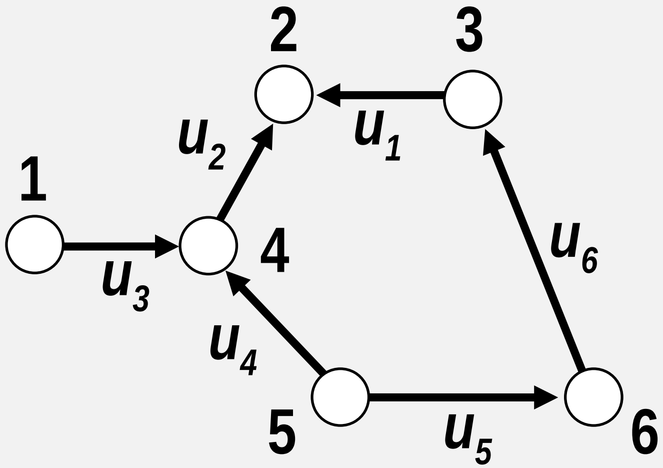
Exemplo: Grafo direcionado

a_{ki} =+1, se a aresta u_k tem origem no vértice i

a_{ki} =-1, se i é o vértice destino da aresta u_k

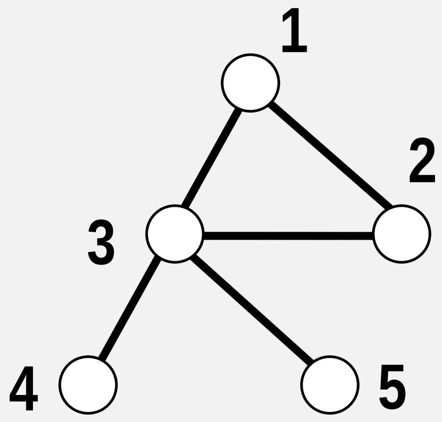
a_{ki} = 0, se a aresta u_k não incide no vértice i

$$
\begin{array}{c}
\\
u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} \;\;1 & \;\;2 & \;\;3 & \;\;4 & \;\;5 & \;\;6 \end{array} \\
\begin{bmatrix}
0 & -1 & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & -1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$



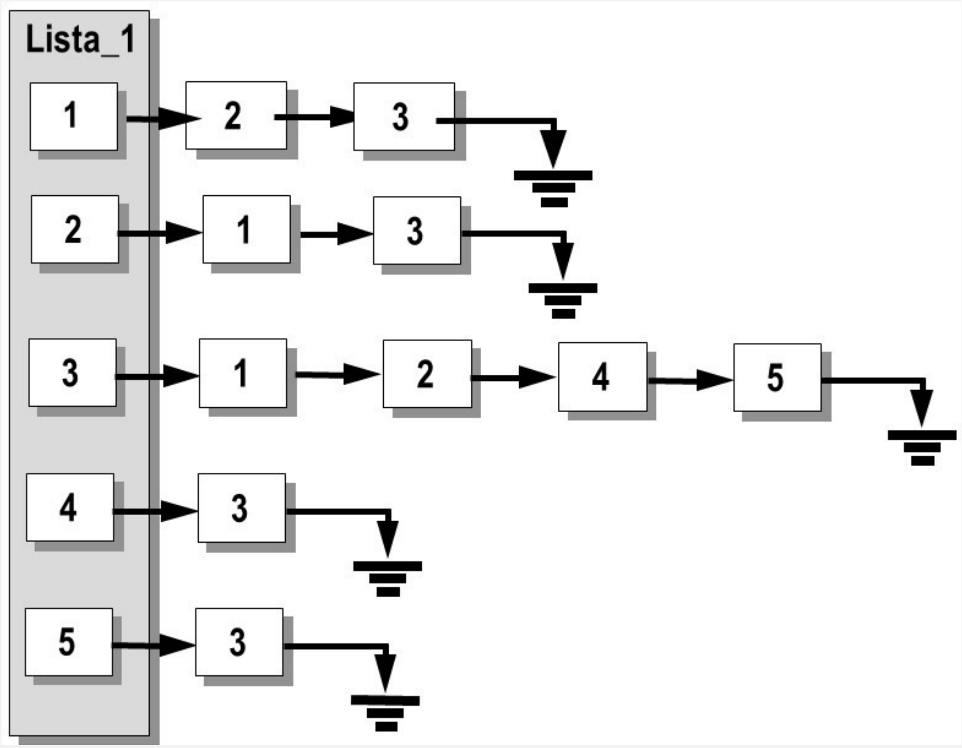
Lista de Adjacência

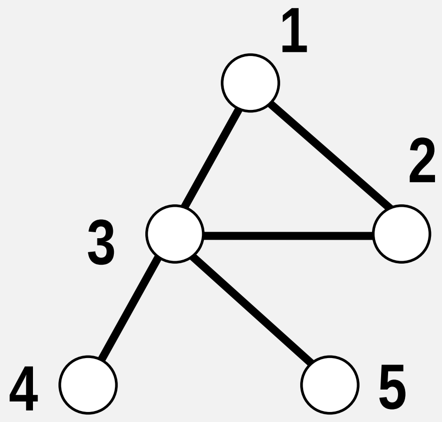
A Lista de Adjacência é composta por um vetor *V* de dimensão *n.* Cada elemento de *V* contém dois campos: a identificação de um vértice e um ponteiro para uma lista encadeada contendo os vizinhos do vértice correspondente.



Lista de Adjacência

A Lista de Adjacência é composta por um vetor *V* de dimensão *n.* Cada elemento de *V* contém dois campos: a identificação de um vértice e um ponteiro para uma lista encadeada contendo os vizinhos do vértice correspondente.





Estrela Direta

A **Representação em Estrela Direta** ocupa ***O***(m+n) posições de memória

São utilizadas:

- duas listas de tamanho m+1 referentes aos **arcos**
- uma lista de tamanho n+1 contendo inteiros.

O k-ésimo elemento da primeira lista de arcos guarda o peso do k-ésimo arco.

A segunda lista contém m pares, cada um identificando o vértice inicial e o vértice final do arco correspondente. A posição m+1 é somente uma posição de referência.

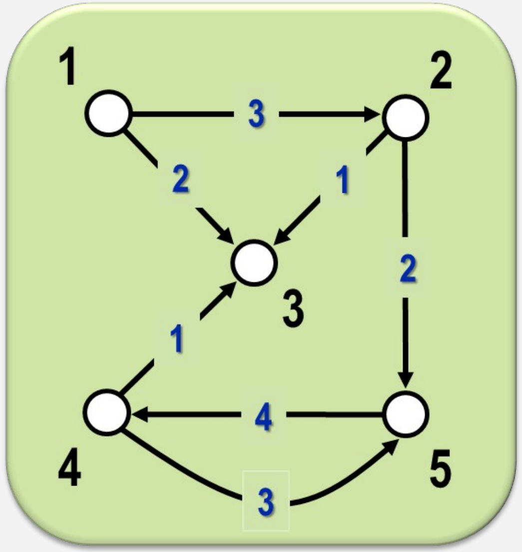
Na terceira lista, chamada ***pont***, o elemento i contém o menor índice da lista de **arcos que saem do vértice *i***.

Caso não existam arcos saindo do vértice i, o elemento *pont*[i] corresponde ao índice do primeiro arco que sai do i+1-ésimo vértice.

Para efeito de consistência da estrutura, pont[1] = 1 e pont[n+1] = m+1.

Estrela Direta

Exemplo



Arcos	Peso do Arco	(início,fim)	pont	Vértice
1				1
2				2
3				3
4				4
5				5
6				
7				
8	-	(-,-)		

Considere que a lista de arcos que saem do vértice *i* está entre os índices *j* e *k*.

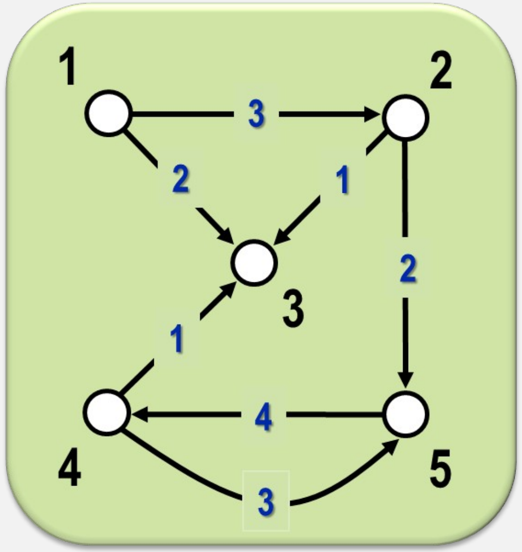
pont[*i*] = *j*
pont[*i*+1] = *k* + 1

Portanto, a lista do arcos que saem do vértice *i* está entre os índices *pont*[*i*] e *pont*[*i*+1] – 1.

Estrela Direta

Estrela Direta

Exemplo



Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

$\underbrace{\hspace{10em}}_{\text{Estrela Direta}}$

Estrela Direta

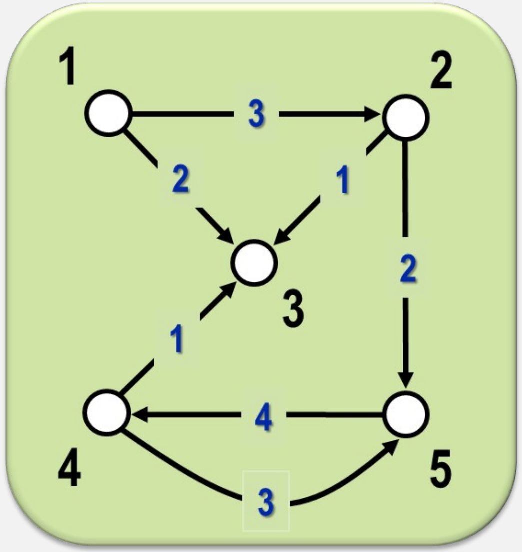
Considere que a lista de arcos que saem do vértice i está entre os índices j e k.

$pont[i] = j$
$pont[i+1] = k + 1$

Portanto, a lista do arcos que saem do vértice i está entre os índices $pont[i]$ e $pont[i+1] - 1$.

Estrela Direta

Exemplo

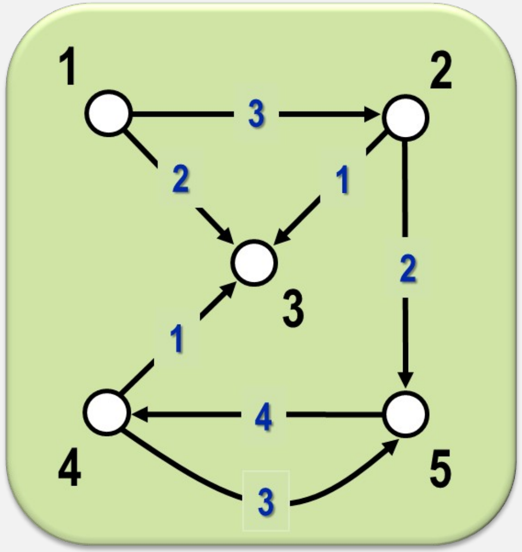


Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

$\underbrace{\hspace{6cm}}_{\text{Estrela Direta}}$

Estrela Direta

Exemplo

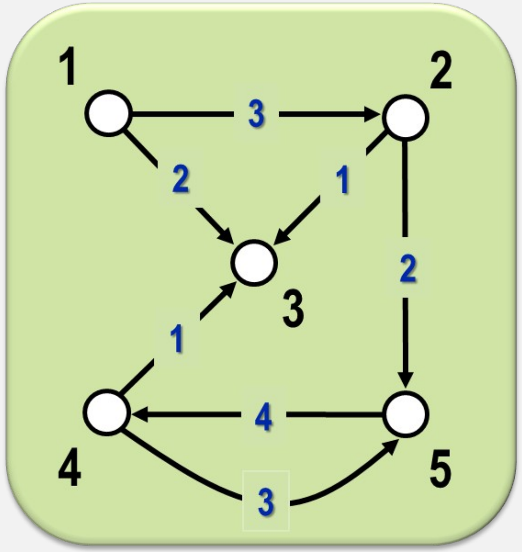


Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

Estrela Direta

Estrela Direta

Exemplo

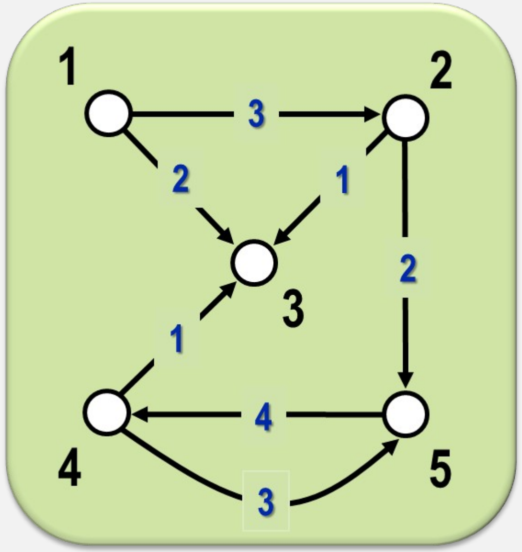


Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

Estrela Direta

Estrela Direta

Exemplo

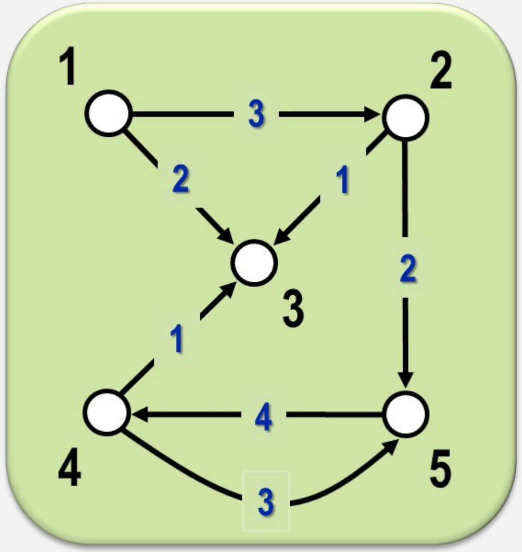


Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

Estrela Direta

Estrela Direta

Exemplo



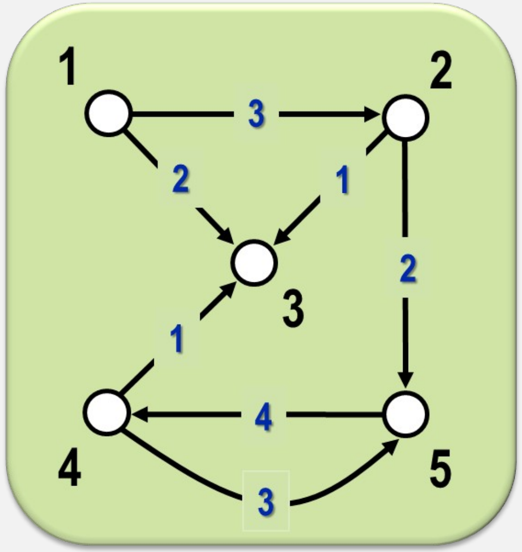
Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

$\underbrace{\hspace{10em}}$

Estrela Direta

Estrela Direta

Exemplo



Arcos	Peso do Arco	(início,fim)	pont	Vértice
1	3	(1,2)	1	1
2	2	(1,3)	3	2
3	1	(2,3)	5	3
4	2	(2,5)	5	4
5	1	(4,3)	7	5
6	3	(4,5)	8	
7	4	(5,4)		
8	-	(-,-)		

Estrela Direta

Estrela Reversa

São utilizadas:

- duas listas de tamanho *m*+1 referentes aos **arcos**
- uma lista de tamanho *n*+1 contendo inteiros.

O *k*-ésimo elemento da primeira lista guarda o peso do *k*-ésimo arco.

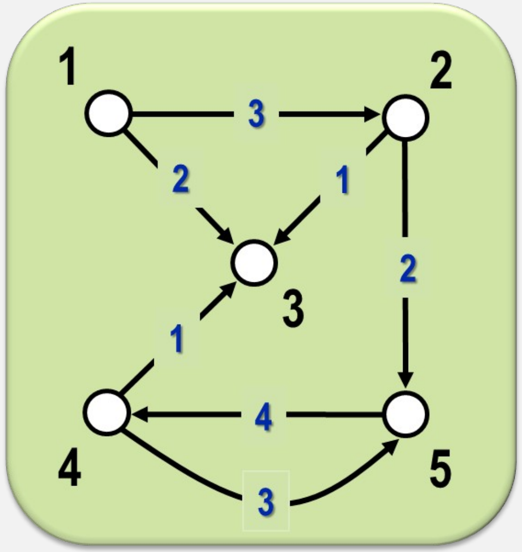
A segunda lista contém *m* pares, cada um identificando o vértice inicial e o vértice final do arco correspondente. A posição *m*+1 é somente uma posição de referência.

Na terceira lista, chamada ***rpont***, o elemento *i* contém o menor índice da lista de **arcos que chegam ao vértice *i***. Caso não existam arcos chegando ao vértice *i*, o elemento *pont*[*i*] corresponde ao índice do primeiro arco que chega ao *i*+1-ésimo vértice.

Para efeito de consistência da estrutura, pont[1] = 1 e pont[*n*+1] = *m*+1.

Estrela Reversa

Exemplo



Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)		1
2	2	(1,3)		2
3	1	(2,3)		3
4	1	(4,3)		4
5	4	(5,4)		5
6	2	(2,5)		
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

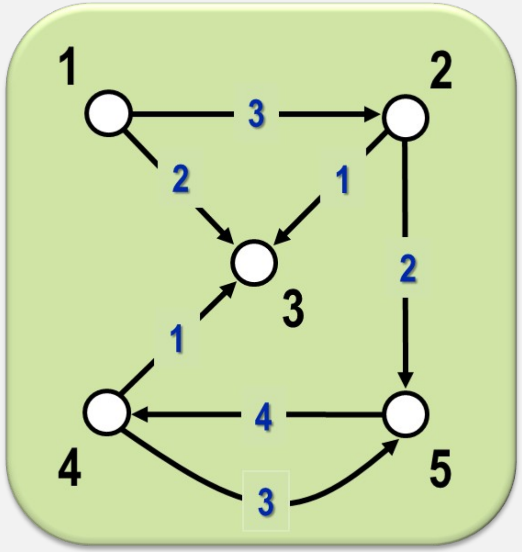
Considere que a lista de arcos que chegam ao vértice *i* está entre os índices *j* e *k*.

rpont[*i*] = *j*
rpont[*i*+1] = *k* + 1

Portanto, a lista do arcos que chegam ao vértice *i* está entre os índices *rpont*[*i*] e *rpont*[*i*+1] – 1.

Estrela Reversa

Exemplo



Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

$\underbrace{\hspace{8em}}_{\text{Estrela Reversa}}$

Estrela Reversa

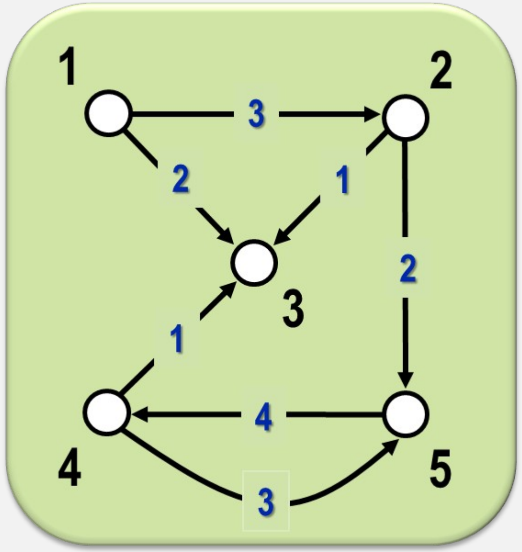
Considere que a lista de arcos que chegam ao vértice *i* está entre os índices *j* e *k*.

rpont[*i*] = *j*
rpont[*i*+1] = *k* + 1

Portanto, a lista do arcos que chegam ao vértice *i* está entre os índices *rpont*[*i*] e *rpont*[*i*+1] – 1.

Estrela Reversa

Exemplo

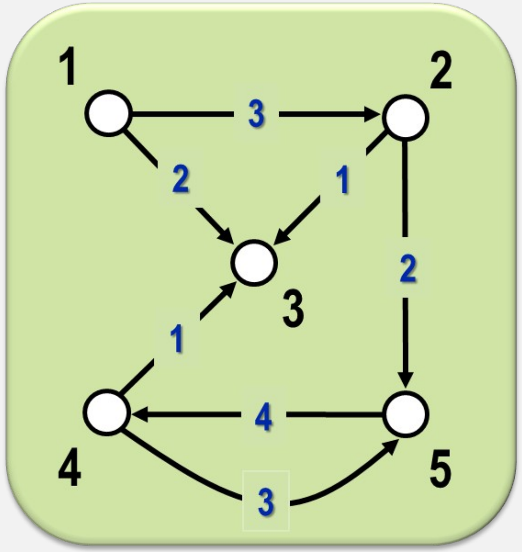


Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Estrela Reversa

Exemplo

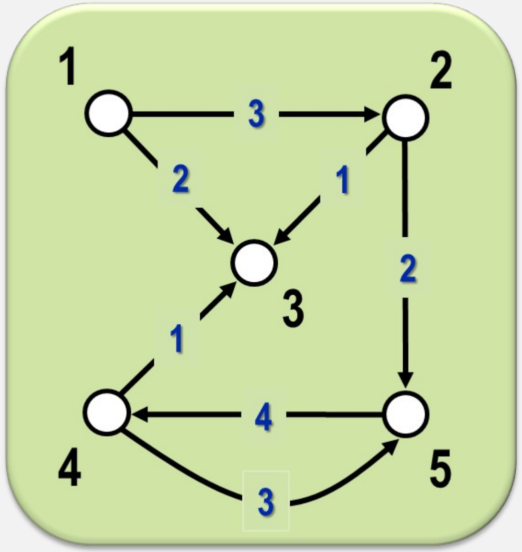


Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Estrela Reversa

Exemplo

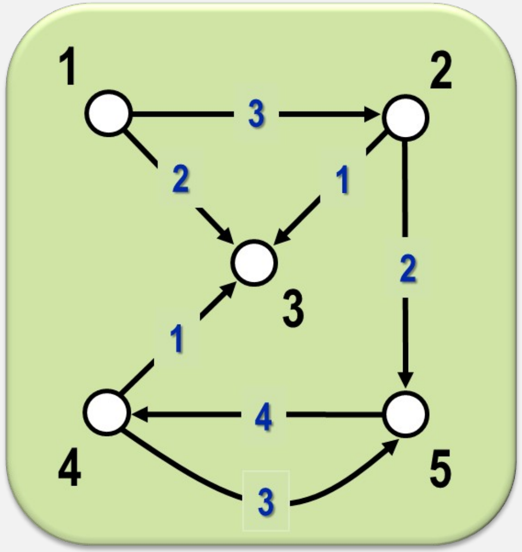


Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Estrela Reversa

Exemplo

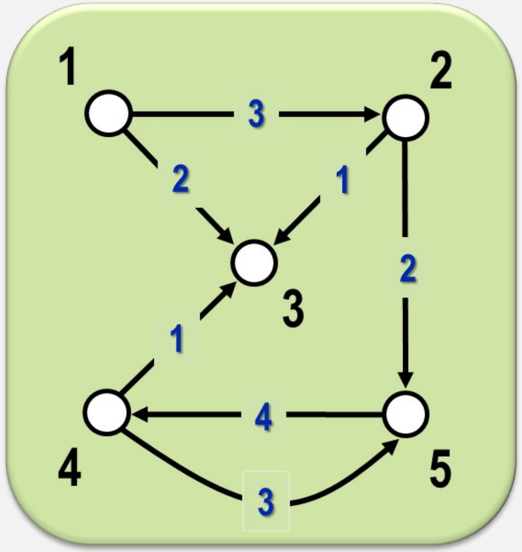


Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	$\boxed{(5,4)}$	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Estrela Reversa

Exemplo

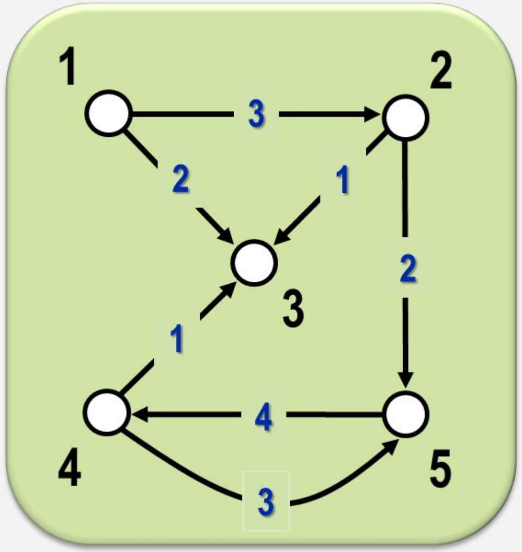


Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Estrela Reversa

Exemplo



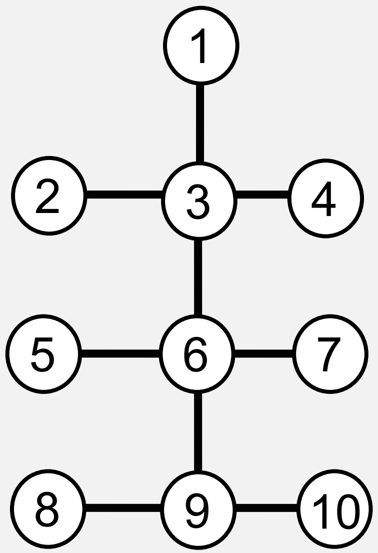
Arcos	Peso do Arco	(início,fim)	rpont	Vértices
1	3	(1,2)	1	1
2	2	(1,3)	1	2
3	1	(2,3)	2	3
4	1	(4,3)	5	4
5	4	(5,4)	6	5
6	2	(2,5)	8	
7	3	(4,5)		
8	-	(-,-)		

Estrela Reversa

Árvores – Código de Prüffer

Considere T = (V_T,E_T) uma árvore rotulada em vértices, onde $|V_T| = n$.

O código de Prüffer para T é uma sequência de n-2 inteiros.



1	2	3	4	5	6	7	8

Código com 8 inteiros

Árvores – Código de Prüffer

Algoritmo:

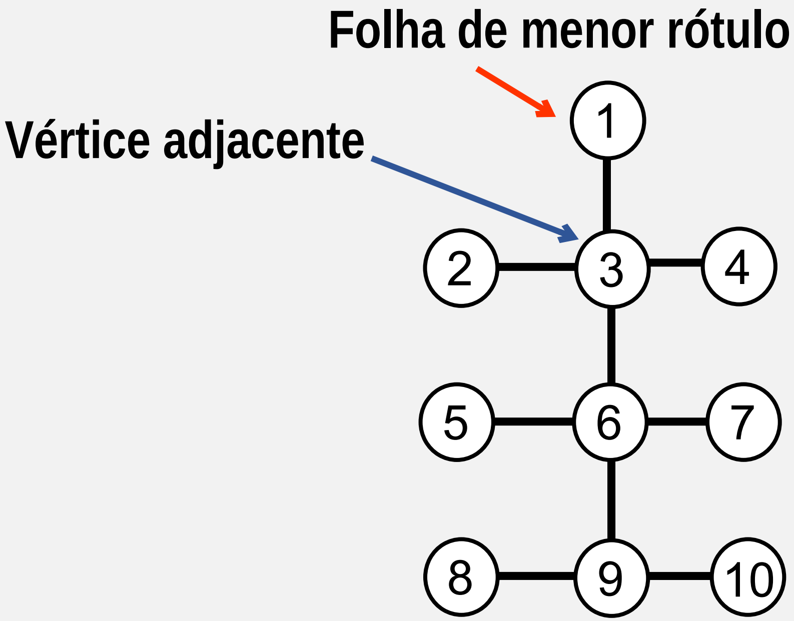
Retira as folhas de menor rótulo da árvore até sobrarem apenas 2 vértices.

No i-ésimo passo, a folha com o menor rótulo é removida.

O i-ésimo elemento da sequência de Prüffer é o rótulo do vértice adjacente a i-ésima folha removida.

Árvores – Código de Prüffer

Exemplo:

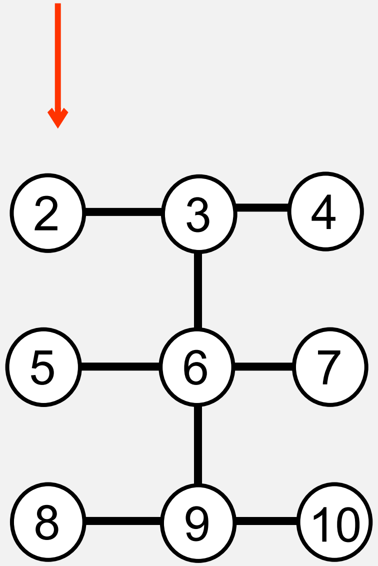


3							
1	2	3	4	5	6	7	8

Árvores – Código de Prüffer

Exemplo:

Folha de menor rótulo

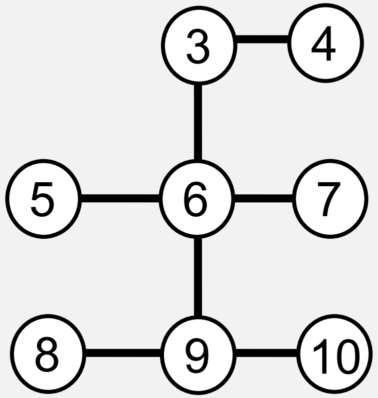


3	3						
1	2	3	4	5	6	7	8

Árvores – Código de Prüffer

Exemplo:

Folha de menor rótulo

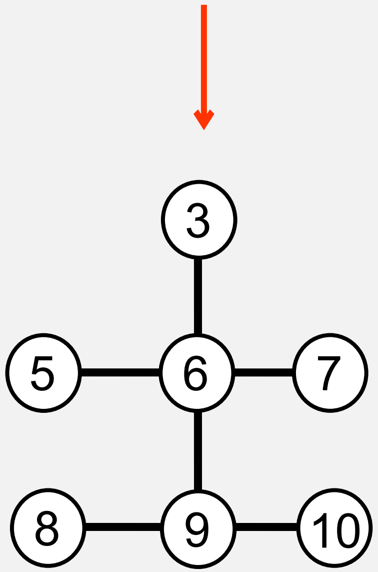


3	3	3					
1	2	3	4	5	6	7	8

Árvores – Código de Prüffer

Exemplo:

Folha de menor rótulo

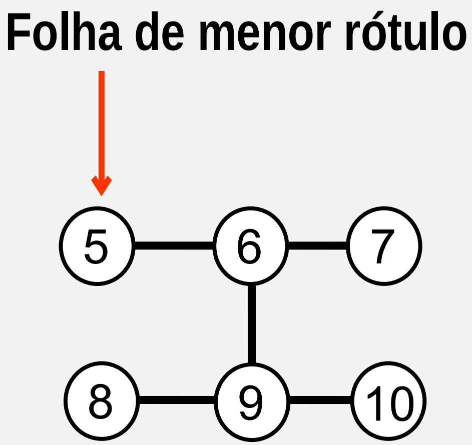


3	3	3	6				
1	2	3	4	5	6	7	8

Árvores – Código de Prüffer

Exemplo:

3	3	3	6	6			
1	2	3	4	5	6	7	8

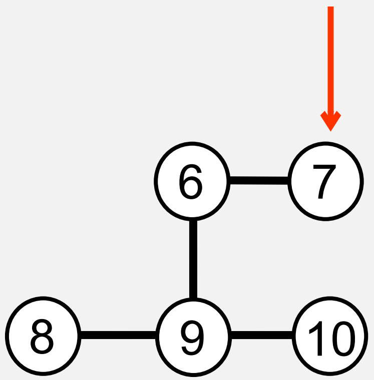


Árvores – Código de Prüffer

Exemplo:

3	3	3	6	6	6		
1	2	3	4	5	6	7	8

Folha de menor rótulo

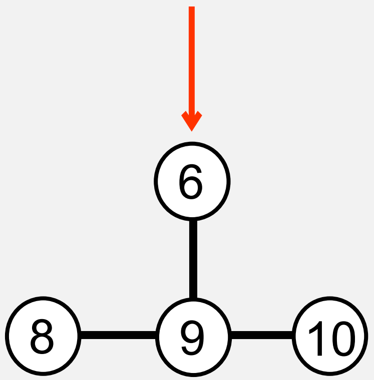


Árvores – Código de Prüffer

Exemplo:

3	3	3	6	6	6	9	
1	2	3	4	5	6	7	8

Folha de menor rótulo

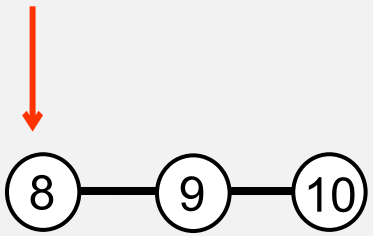


Árvores – Código de Prüffer

Exemplo:

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Folha de menor rótulo

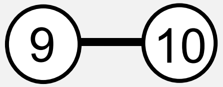


Árvores – Código de Prüffer

Exemplo:

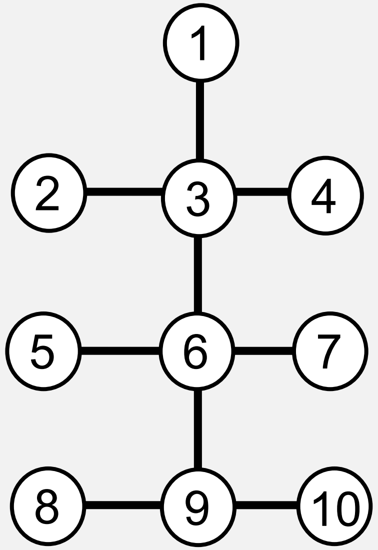
3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Até restar 2 vértices



Árvores – Código de Prüffer

Exemplo:



3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Obs. O grau de cada vértice da árvore é igual ao número de vezes que o vértice aparece no código de Prüffer + 1.

Árvores – Código de Prüffer

Exemplo:

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

O código nos dá de imediato duas informações:

O número de vértices da árvore

e

O grau de cada vértice

Árvores – Código de Prüffer

Algoritmo para recuperar a árvore a partir do código de Prüffer

Considere $p_1, p_2, \ldots, p_{n-2} \in \{1,2,\ldots,n\}^{n-2}$ um código de Prüffer.

1. Calcular o grau de cada vértice e colocar na lista *L*. (O grau de cada vértice da árvore é igual ao número de vezes que ele aparece no código de Prüffer + 1.) Iniciar i com 1.
2. Encontre na lista *L* o nó *v* de grau 1 com o menor rótulo. (v,p_i) é uma aresta da árvore.
3. Decremente os graus de *v* e p_i; Incremente *i*
4. Repita os passos 2 e 3 até todos os nós terem grau 0, a menos de 2 com grau 1. Estes dois formam a última aresta da árvore

Árvores – Código de Prüffer

Exemplo:

O grau de cada vértice da árvore é igual ao número de vezes que ele aparece no código de Prüffer + 1. Iniciar *i* com 1.

i = 1

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer

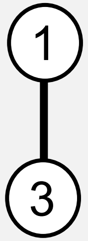
1	1	4	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore

Árvores – Código de Prüffer

Exemplo:

2. Encontre o nó *v* de grau 1 com o menor rótulo. (v,p_i) é uma aresta da árvore.
3. Decremente os graus de *v* e p_i; Incremente *i*



i = 1

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer

1	1	4	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore

v = 1

graus decrementados

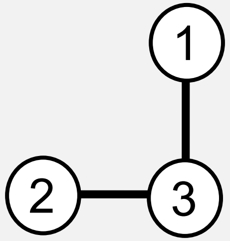
0	1	3	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Árvores – Código de Prüffer

Exemplo:

2. Encontre o nó *v* de grau 1 com o menor rótulo. (v,p_i) é uma aresta da árvore.

3. Decremente os graus de *v* e p_i; Incremente *i*



i = 2

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer

0	1	3	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore

v = 2

graus decrementados

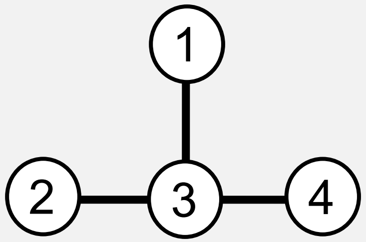
0	0	2	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Árvores – Código de Prüffer

Exemplo:

2. Encontre o nó *v* de grau 1 com o menor rótulo. (v,p_i) é uma aresta da árvore.

3. Decremente os graus de *v* e p_i; Incremente *i*



i = 3

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer

0	0	2	1	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore

v = 4

graus decrementados

0	0	1	0	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Árvores – Código de Prüffer

Exemplo:

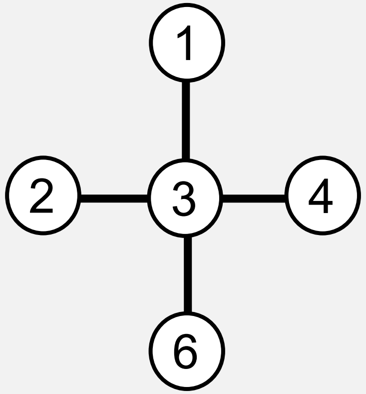
2. Encontre o nó *v* de grau 1 com o menor rótulo. (v,p_i) é uma aresta da árvore.

3. Decremente os graus de *v* e p_i; Incremente *i*

$i = 4$

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer



$v = 3$

graus decrementados

0	0	1	0	1	4	1	1	3	1
1	2	3	4	5	6	7	8	9	10

0	0	0	0	1	3	1	1	3	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore

Árvores – Código de Prüffer

Exemplo:

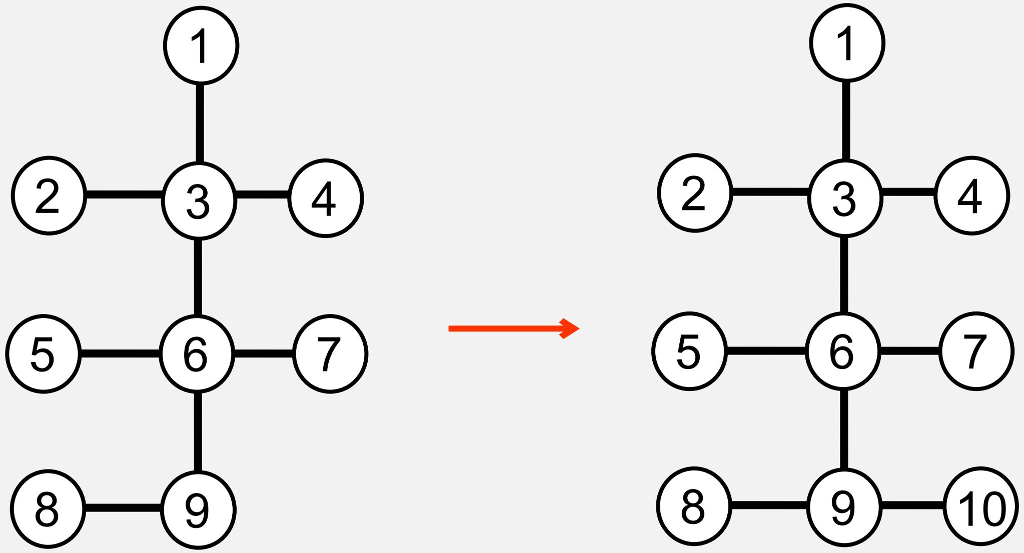
4. Repita os passos 2 e 3 até todos os nós terem grau 0, a menos de 2 com grau 1. Estes dois formam a última aresta da árvore

3	3	3	6	6	6	9	9
1	2	3	4	5	6	7	8

Código de Prüffer

0	0	0	0	0	0	0	0	1	1
1	2	3	4	5	6	7	8	9	10

Graus dos vértices da árvore



Árvores – Código de Prüffer

Uma implementação eficiente deste algoritmo utiliza uma heap para manter os nós com grau 1.

O algoritmo tem complexidade $O(n\log n)$

Estruturas de Dados

Pontos a Ponderar na Escolha da Estrutura

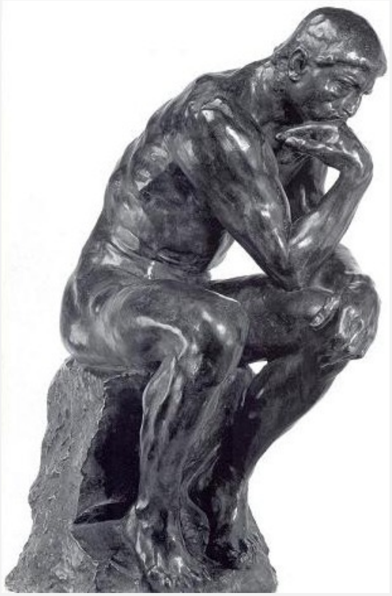
1. Tamanho do grafo

n = 100

10000 entradas na matriz de adjacência

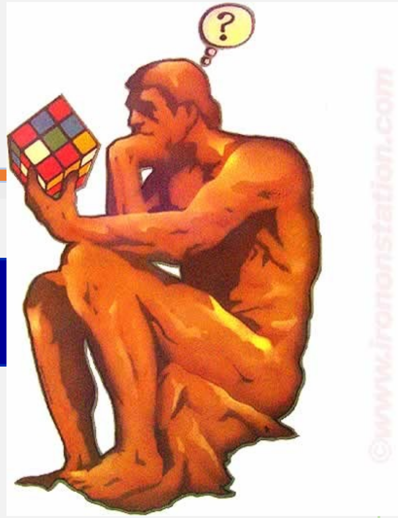
n = 10000

100.000.000 entradas na matriz de adjacência



Matrizes de adjacência são uma escolha adequada para grafos pequenos

Estruturas de Dados



Pontos a Ponderar na Escolha da Estrutura

2. Densidade do grafo

Grafos densos fazem com que a complexidade em espaço das listas de adjacência seja $\theta(n^2)$

Matrizes de adjacência são uma escolha adequada para grafos muito densos. Estrela Direta e Reversa é adequada tanto para grafo denso como para grafo esparso.

Estruturas de Dados

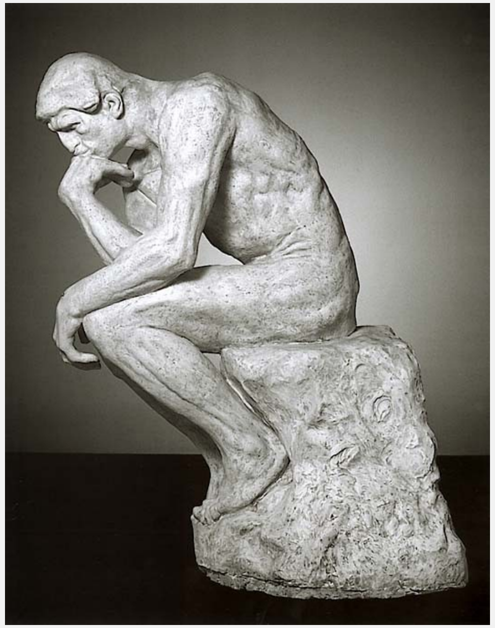
Pontos a Ponderar na Escolha da Estrutura

3. Algoritmos que serão implementados

Alguns algoritmos são mais fáceis de implementar com matrizes de adjacência e outros com lista de adjacência

4. O grafo será constantemente modificado?

Se inserções e remoções serão realizadas repetidas vezes, a matriz de adjacência é a escolha mais adequada. Se os atributos de vértices e arestas serão frequentemente modificados, a melhor opção, uma boa alternativa é trabalhar com campos extra para tais atributos nas matrizes de adjacência.



Exercícios

(1)Crie um programa para criação e manipulação de um grafo simples, incluindo as funções de inserir vértice, excluir vértice e imprimir grafo. Execute a instância do primeiro exemplo desta aula.

(2)Dada uma representação de um grafo não direcionado por matriz de adjacência, faça um algoritmo que represente o grafo em lista de adjacência.

(3)Dada uma representação de um grafo direcionado por matriz de adjacência, faça um algoritmo que represente o grafo em matriz de incidência.

(4)Dada uma representação de um grafo direcionado por matriz de adjacência, faça um algoritmo que represente o grafo em estrela direta e outro em estrela reversa.

(5)Dada uma representação de uma árvore por matriz de adjacência, faça um algoritmo que produza o código de Prüffer.

Analise a complexidade dos algoritmos das questões 2 a 5.