Teoria da Computação

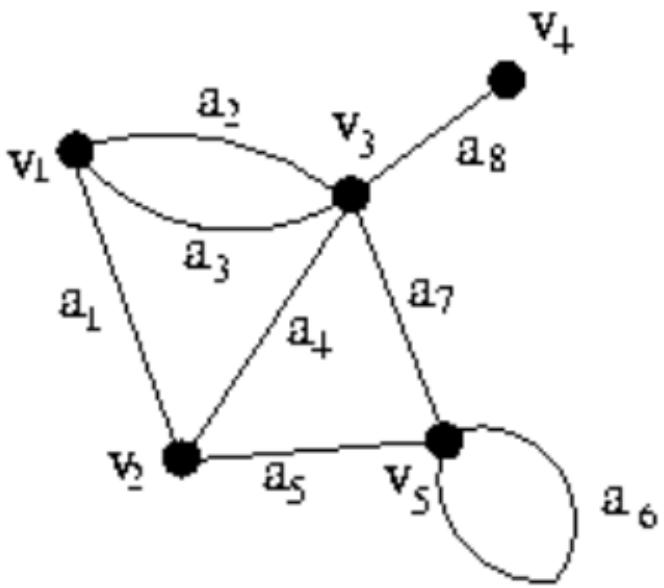
Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

19 de Junho de 2024

Algoritmos em Grafos

- Um grafo é uma estrutura definida por G = (V,E);
- Conjunto não-vazio de vértices;
- Conjunto de elementos denominados arestas;
- Uma aresta é representada por:
  - (vi, vj);



$V = \{v1, v2, v3, v4, v5\}$

$E = \{a1, a2, a3, a4, a5, a6, a7, a8\}$

Sendo que $a1 = (v1,v2)$, $a2 = (v1,v3)$, $a3 = (v1,v3)$, $a4 = (v2,v3)$, $a5 = (v2,v5)$, $a6 = (v5,v5)$, $a7 = (v3,v5)$, $a8 = (v3,v4)$.

Grafos

- Grafo G = (V,E);

$$n = |V| = \text{número de vértices}$$

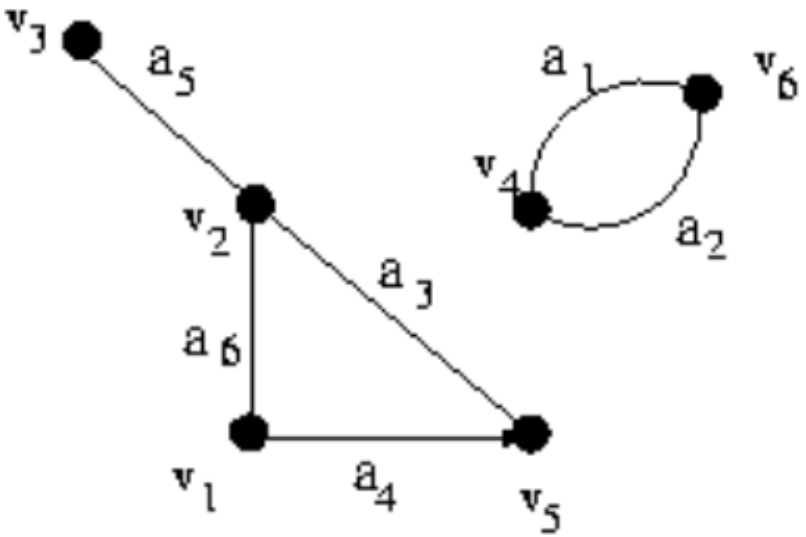
$$m = |E| = \text{número de arestas}$$

Grafos - Representação Computacional

- Matriz de incidência;
- Matriz de adjacência;
- Lista de adjacência;
- Conjunto;

Matriz de incidência

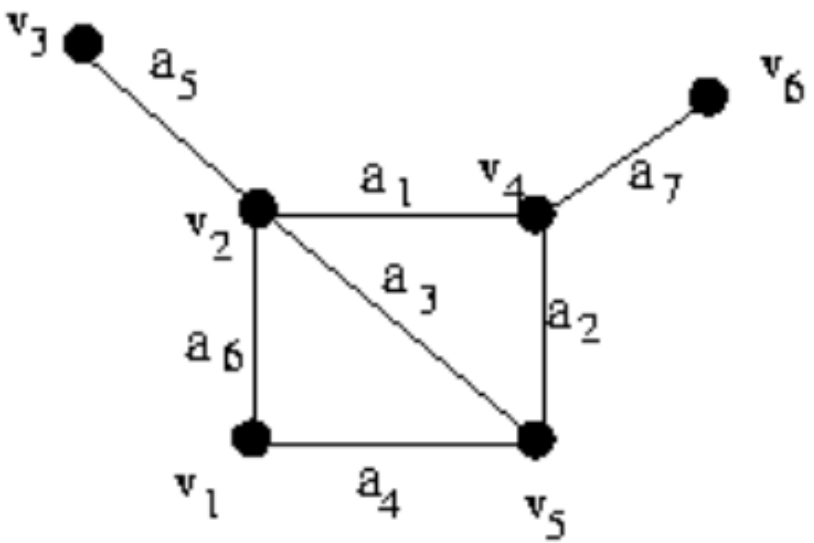
- Matriz n x m;
- Cada elemento m_{ij}:
  - $m_{i_j} = 1$ se a aresta a_j é incidente ao vértice a_i;
  - $m_{i_j} = 0$ caso contrário.



	a_1	a_2	a_3	a_4	a_5	a_6
v_1	0	0	0	1	0	1
v_2	0	0	1	0	1	1
v_3	0	0	0	0	1	0
v_4	1	1	0	0	0	0
v_5	0	0	1	1	0	0
v_6	1	1	0	0	0	0

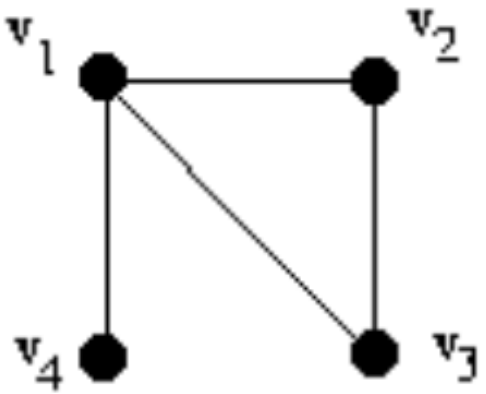
Matriz de adjacência

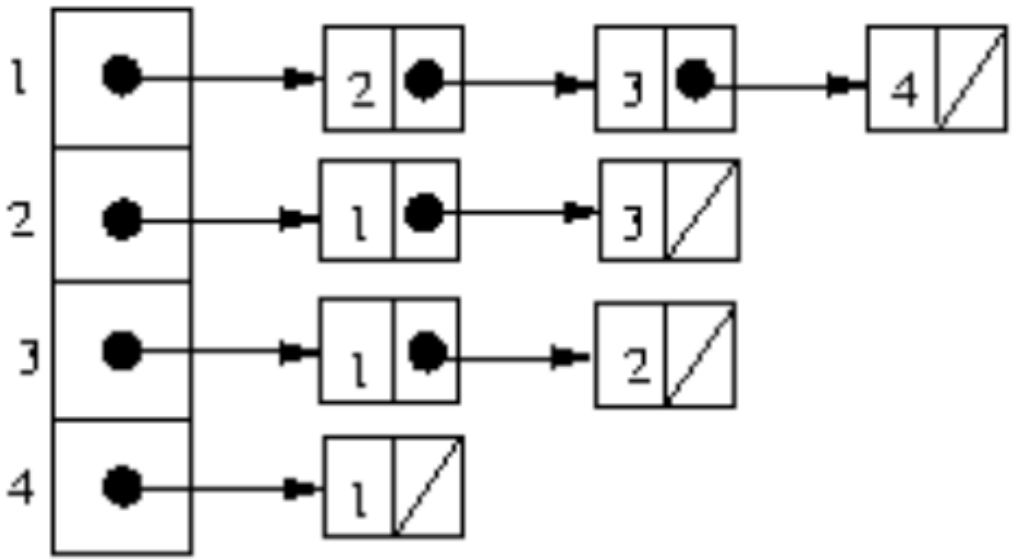
- Matriz de n x n;
- cada elemento e_{jk}:
  - $e_{jk} = 1$ se os vértices v_j e v_k são ligados por uma aresta;
  - $e_{jk} = 0$ caso contrário.



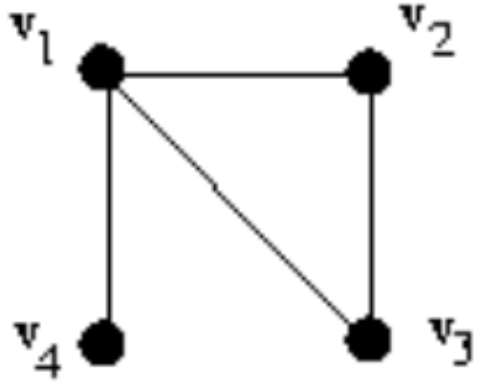
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	1	0
v_2	1	0	1	1	1	0
v_3	0	1	0	0	0	0
v_4	0	1	0	0	1	1
v_5	1	1	0	1	0	0
v_6	0	0	0	1	0	0

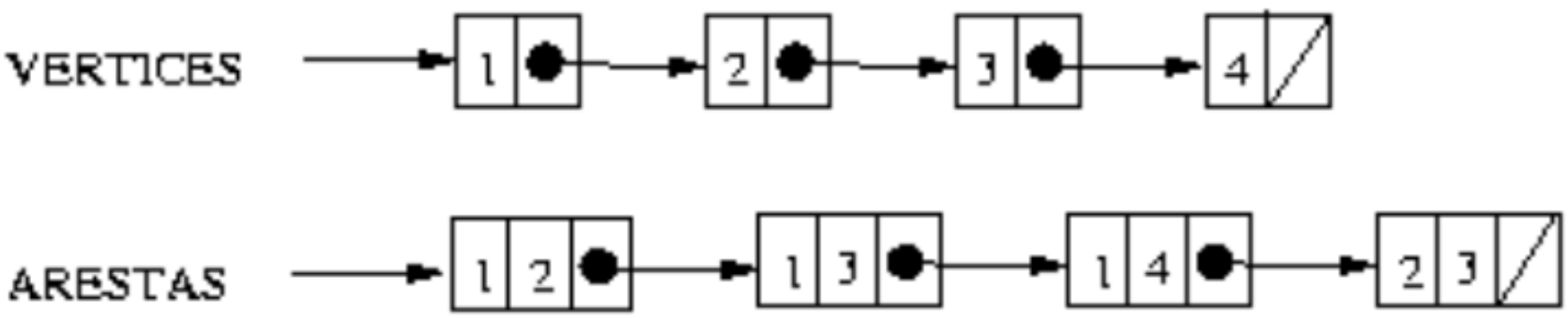
Lista de adjacência





Conjunto





Análise de complexidade - Notação assintótica

- Lidar com duas variáveis;
- Relacionadas ao tamanho da entrada de dados;

Análise de complexidade - Notação assintótica

- Lidar com duas variáveis;
- Relacionadas ao tamanho da entrada de dados:
  - **n**: Número de vértices;
  - **m**: Número de arestas.

Complexidade

	Mat. Inc.	Mat. Adj.	Lista Adj.	Conjuntos
Memória (complexidade de espaço)	*1.1*	*1.2*	*1.3*	*1.4*
Algoritmo para Buscar todos os adjacentes de v_i	*2.1*	*2.2*	*2.3*	*2.4*
Algoritmo para Conferir adjacência de v_i e v_j	*3.1*	*3.2*	*3.3*	*3.4*
Algoritmo para Visitar todas as arestas	*4.1*	*4.2*	*4.3*	*4.4*
Algoritmo para Calcular grau de um vértice	*5.1*	*5.2*	*5.3*	*5.4*

Memória (complexidade de espaço)

	Mat. Inc.	**Mat. Adj.**	**Lista Adj.**	**Conjuntos**
Memória	$O(mn)$	$O(n^2)$	$O(m+n)$	$O(m+n)$

- **Matriz de Incidência (Mat. Inc.)**: $O(mn)$ onde m é o número de arestas e n é o número de vértices.
- **Matriz de Adjacência (Mat. Adj.)**: $O(n^2)$ pois cada par de vértices é considerado.
- **Lista Adjacência (Lista Adj.)**: $O(m+n)$ pois armazena listas de adjacência.
- **Conjuntos**: $O(m+n)$ pois armazena conjuntos de arestas e vértices.

Buscar todos os adjacentes de v_i

	Mat. Inc.	**Mat. Adj.**	**Lista Adj.**	**Conjuntos**
Buscar adjacentes	$O(mn)^*$	$O(n)$	$O(n)^*$	$O(m)$

- **Matriz de Incidência (Mat. Inc.)**: $O(mn)$ no pior caso.
- **Matriz de Adjacência (Mat. Adj.)**: $O(n)$ pois percorre a linha do vértice v_i.
- **Lista Adjacência (Lista Adj.)**: $O(n)$ no pior caso.
- **Conjuntos**: $O(m)$ pois verifica todas as arestas.

Conferir adjacência de v_i e v_j

	Mat. Inc.	**Mat. Adj.**	**Lista Adj.**	**Conjuntos**
Conferir adjacência	$O(m)^*$	$O(1)$	$O(n)^*$	$O(m)^*$

- **Matriz de Incidência (Mat. Inc.)**: $O(m)$ no pior caso.
- **Matriz de Adjacência (Mat. Adj.)**: $O(1)$ pois acessa diretamente a entrada na matriz.
- **Lista Adjacência (Lista Adj.)**: $O(n)$ no pior caso.
- **Conjuntos**: $O(m)$ no pior caso.

Visitar todas as arestas

	Mat. Inc.	**Mat. Adj.**	**Lista Adj.**	**Conjuntos**
Visitar arestas	$O(mn)$	$O(n^2)$	$O(m+n)$	$O(m)$

- **Matriz de Incidência (Mat. Inc.)**: $O(mn)$ pois percorre toda a matriz.
- **Matriz de Adjacência (Mat. Adj.)**: $O(n^2)$ pois percorre todas as entradas.
- **Lista Adjacência (Lista Adj.)**: $O(m+n)$ pois visita cada lista de adjacência.
- **Conjuntos**: $O(m)$ pois visita cada aresta uma vez.

Calcular grau de um vértice

	Mat. Inc.	**Mat. Adj.**	**Lista Adj.**	**Conjuntos**
Calcular grau	$O(m)$	$O(n)$	$O(n)^{**}$	$O(m)$

- **Matriz de Incidência (Mat. Inc.)**: $O(m)$ pois verifica todas as arestas incidentes.
- **Matriz de Adjacência (Mat. Adj.)**: $O(n)$ pois conta as entradas na linha do vértice.
- **Lista Adjacência (Lista Adj.)**: $O(n)$ no pior caso para grafos não orientados.
- **Conjuntos**: $O(m)$ pois verifica todas as arestas.

Complexidade

	Mat. Inc.	Mat. Adj.	Lista Adj.	Conjuntos
Memória (complexidade de espaço)	$O(mn)$	$O(n^2))$	$O(m+n)$	$O(m+n)$
Algoritmo para Buscar todos os adjacentes de v_i	$O(mn)$	$O(n)$	$O(n)$	$O(m)$
Algoritmo para Conferir adjacência de v_i e v_j	$O(m)$	$O(1)$	$O(n)$	$O(m)$
Algoritmo para Visitar todas as arestas	$O(mn)$	$O(n^2)$	$O(m+n)$	$O(m)$
Algoritmo para Calcular grau de um vértice	$O(m)$	$O(n)$	$O(n)$	$O(m)$

Busca em grafos

- Busca em profundidade;
- Busca em largura;

Busca em grafos

- É possível montar sua equação de recorrência?
- É possível analisar a complexidade linha por linha?

Busca em grafos

- É possível montar sua equação de recorrência?
- É possível analisar a complexidade linha por linha?
  - Tempo para visitar todos os vértices;
  - Tempo para processar todas as arestas adjacentes a cada vértice;
  - O(n + m);

Busca em grafos

- É possível montar sua equação de recorrência?
- É possível analisar a complexidade linha por linha?
- A complexidade depende da estrutura de dados utilizada?

Busca em grafos

- É possível montar sua equação de recorrência?
- É possível analisar a complexidade linha por linha?
- A complexidade depende da estrutura de dados utilizada?
  - Lista de Adjacência:
    - Espaço: O(n+m)
    - Tempo: O(n+m)
  - Matriz de Adjacência:
    - Espaço: $O(n^2)$
    - Tempo: $O(n^2)$
  - Matriz de Incidência:
    - Espaço: O(nm)
    - Tempo: O(nm)

Atividade - Parte 1

- Montar grupos de até 3 pessoas e explicar os problemas a seguir, discutindo também a complexidade computacional de cada um:
  - Colore um grafo;
  - Árvore Geradora Mínima;
  - Componentes Fortemente Conectados;
  - Identificar articulações no grafo.

Atividade - Parte 2

- Escreva um algoritmo e de sua complexidade computacional para:
  - Colore um grafo;
  - Árvore Geradora Mínima;
  - Componentes Fortemente Conectados;
  - Identificar articulações no grafo.

Bibliografia Básica

- LEWIS, H. R.; PAPADIMITRIOU, C. H. **Elementos de Teoria da Computação**. 2 ed. Porto Alegre: Bookman, 2000.
- VIEIRA, N. J. **Introdução aos Fundamentos da Computação**. Editora Pioneira Thomson Learning, 2006.
- DIVERIO, T. A.; MENEZES, P. B. **Teoria da Computação: Máquinas Universais e Computabilidade**. Série Livros Didáticos Número 5, Instituto de Informática da UFRGS, Editora Sagra Luzzato, 1 ed. 1999.

Obrigado!
Dúvidas?

Guilherme Henrique de Souza Nakahata

guilhermenakahata@gmail.com

https://github.com/GuilhermeNakahata/UNESPAR-2024