

Grafos – Introdução

Fernando Lobo

Algoritmos e Estrutura de Dados

1 / 24

Grafos – Introdução

- Estrutura de dados fundamental em Informática, tal como listas e árvores.
- Há muitos algoritmos interessantes sobre grafos.
- Um grafo é um conceito abstracto (matemático) com muitas aplicações práticas.
- Já devem ter aprendido umas coisas sobre grafos em Matemática Discreta.

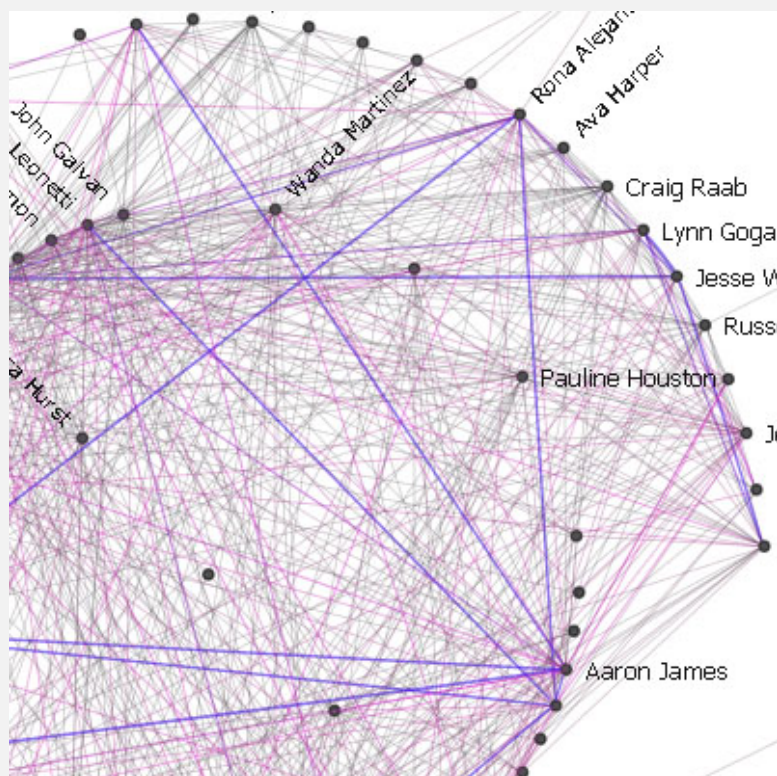
2 / 24

Um grafo é uma rede

- Rede de computadores.
- Rede de estradas.
- Rede eléctrica.
- Rede social.
- A web.
- etc.

3 / 24

Facebook



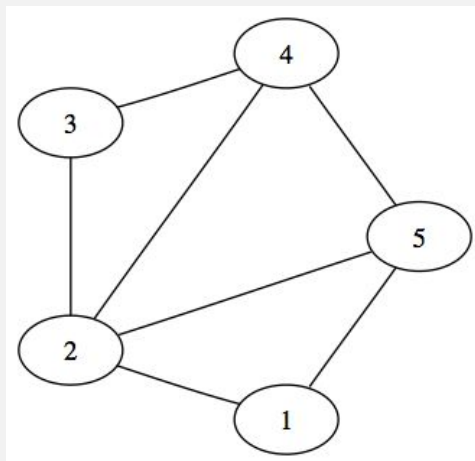
4 / 24

Grafos

- Um grafo pode ser orientado ou não orientado (em inglês, directed ou undirected).
- Notação: $G = (V, E)$, em que V é o conjunto de vértices ou nós (nodes) e E é o conjunto de arcos (edges).

5 / 24

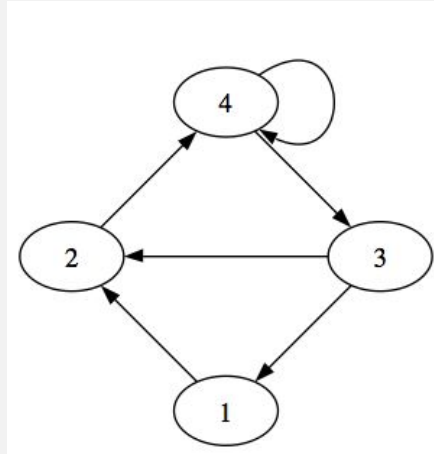
Exemplo de um grafo não orientado



- O sentido dos arcos não interessa: $(u, v) = (v, u)$.
- $V = \{1, 2, 3, 4, 5\}$
- $E = \{(1, 2), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (4, 5)\}$

6 / 24

Exemplo de um grafo orientado



- O sentido dos arcos interessa: $(u, v) \neq (v, u)$.
- $V = \{1, 2, 3, 4\}$
- $E = \{(1, 2), (2, 4), (3, 1), (3, 2), (4, 3), (4, 4)\}$

7 / 24

Representação

Há duas representações principais:

- 1 Matriz de adjacência.
- 2 Array de listas de adjacências.

8 / 24

Matriz de adjacência

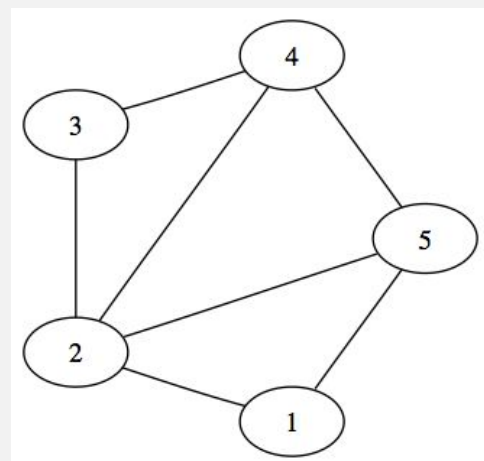
- Constrói-se uma matriz A de $|V| \cdot |V|$ em que:

$$A_{ij} = \begin{cases} 1 & , \text{ se } (i,j) \in E \\ 0 & , \text{ caso contrário} \end{cases}$$

9 / 24

Grafo não orientado

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

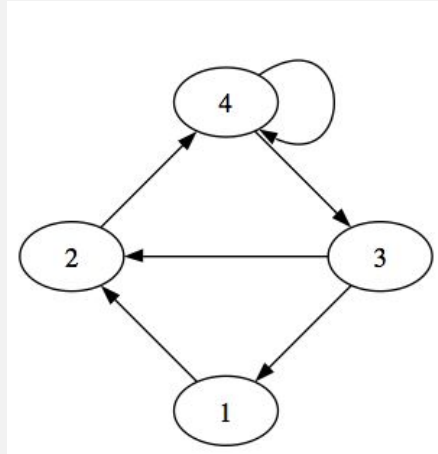


Matriz é simétrica: $A[i,j] = A[j,i]$.

10 / 24

Grafo orientado

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	1	1	0	0
4	0	0	1	1

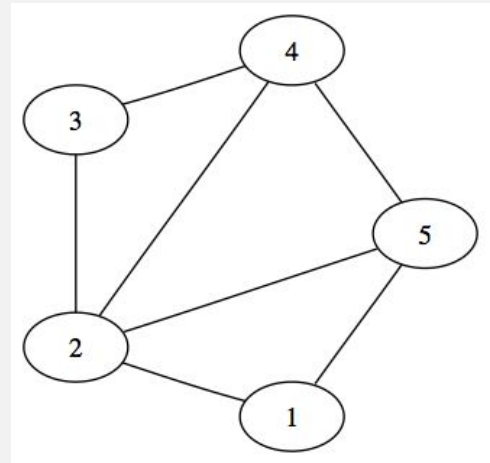
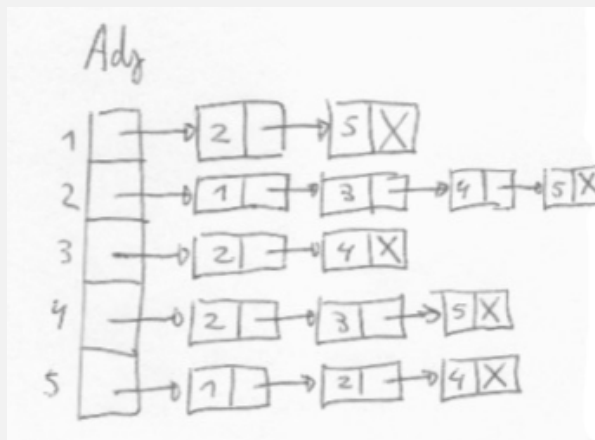


Matriz não é simétrica.

Array de listas de adjacência

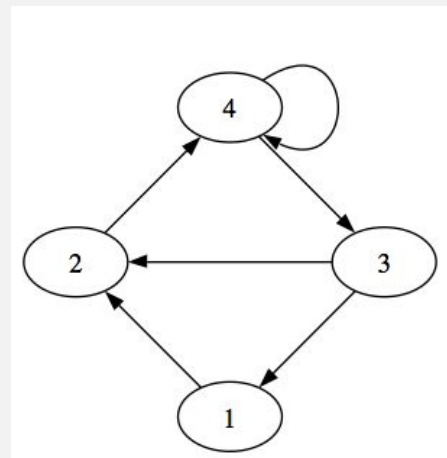
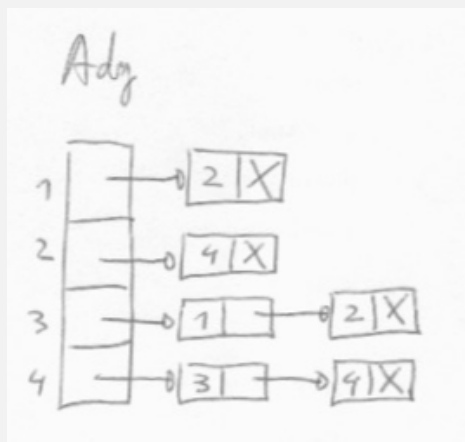
- O grafo é representado por um array de listas, uma lista para cada vértice.
- A lista do vértice v contém os vértices adjacentes a v .

Grafo não orientado



13 / 24

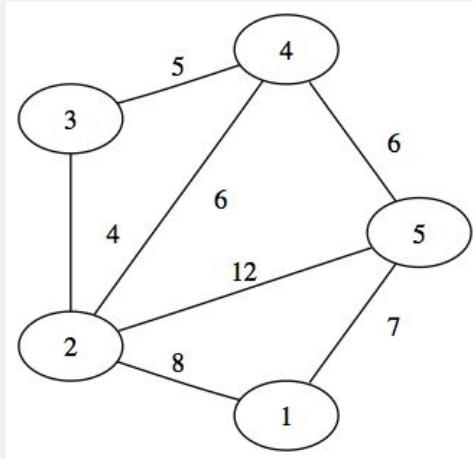
Grafo orientado



14 / 24

Grafos com pesos

- Por vezes, os arcos têm pesos associados. Podem representar custos, distâncias, etc ...

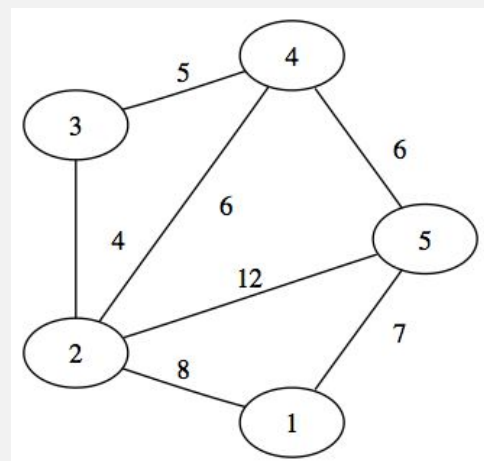


15 / 24

Representação com matriz de adjacência

- Na matriz de adjacência guardamos o peso em vez do bit. Usa-se um número diferente de qualquer peso possível para denotar a ausência do arco.

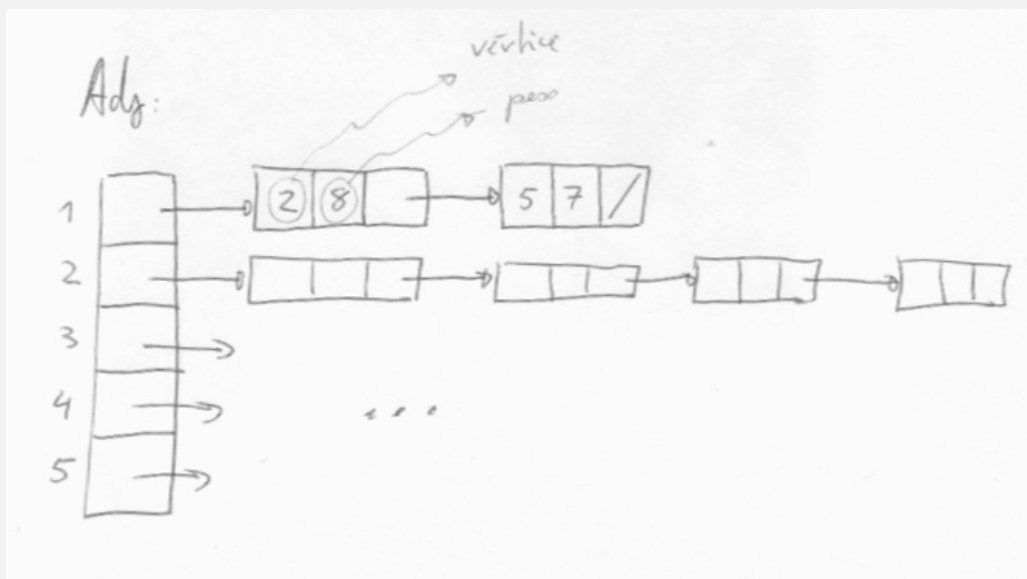
	1	2	3	4	5
1	-1	8	-1	-1	7
2	8	-1	4	6	12
3	-1	4	-1	5	-1
4	-1	6	5	-1	6
5	7	12	-1	6	-1



16 / 24

Representação com array de listas de adjacência

- Nas listas de adjacência guardamos em cada elemento da lista o nó e o peso correspondente.



17 / 24

Matriz de adjacência versus Array de listas

- Qual a melhor representação?
- Resposta: Depende.
 - ▶ Matriz é melhor se o grafo for denso. Isto é, se o número de arcos = $\Theta(|V|^2)$.
 - ▶ Array de listas é melhor se o grafo não for denso (i.e., for esparso) → porque a maioria das entradas da matriz estão vazias.

18 / 24

Complexidade

- Temos visto a complexidade de algoritmos em termos de um só parâmetro N .
- Nos grafos, a complexidade é dada em termos de dois parâmetros: $|V|$ e $|E|$.
- Na notação assintótica é costume usar V e E em vez de $|V|$ e $|E|$. Trata-se de um abuso de notação, mas torna as expressões mais simples.

Complexidade espacial

- Matriz: $\Theta(V^2)$
- Array de listas: $\Theta(V + E)$

Complexidade temporal

- Depende das operações
- Vejamos duas operações essenciais:

op1: determinar se $(u, v) \in E$.

op2: listar todos os nós adjacentes de um determinado nó u .

21 / 24

Complexidade temporal

	op1	op2
Matriz	$\Theta(1)$	$\Theta(V)$
Array de listas	$\Theta(\text{grau}(u))$	$\Theta(\text{grau}(u))$

- $\text{grau}(u)$ é o número de nós adjacentes a u . Em inglês, $\text{degree}(u)$.
- Matriz é melhor para op1. Array de listas é melhor para op2.

22 / 24

Na prática

- Grafos densos → Matriz de adjacência.
- Grafos esparsos → Array de listas.
- Na maioria das aplicações, os grafos são esparsos.
- Usaremos a representação de array de listas na maioria dos algoritmos que vamos ver.

Uma nota

- Nos exemplos coloquei números inteiros para identificar nós.
- Na prática os nós podem ter nomes.
- É necessário uma tabela que mapeia nomes de nós para os números.