AULA 15

Prof. Mathias

Algoritmos de busca

Análise de Algoritmos

Agenda

- Aula anterior
- Introdução
- Busca em largura
- Busca em profundidade
- Exercícios
- Próxima aula

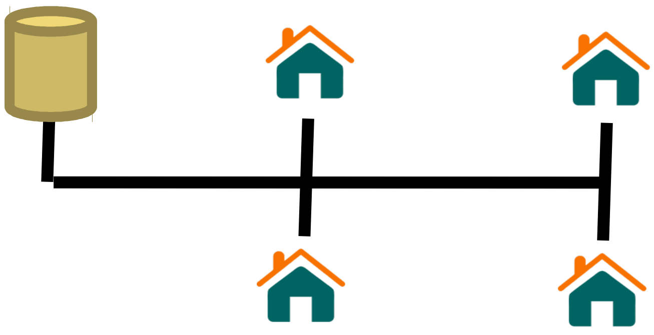
Aula Anterior

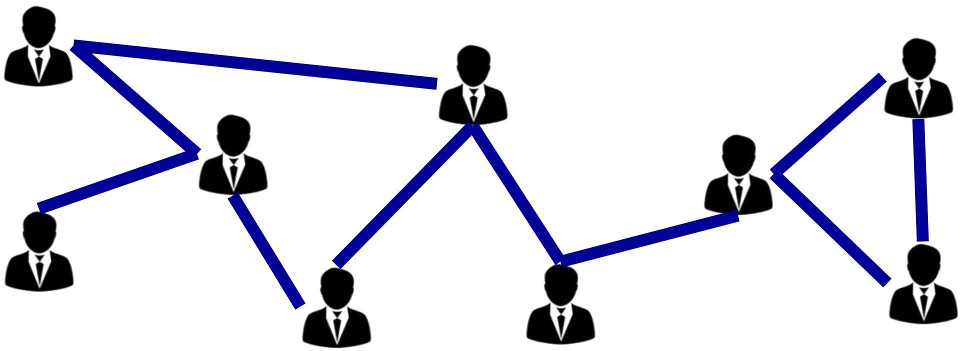
- Busca linear estática
- Busca Sequêncial estática
- Busca binária estática

Agenda

- Aula anterior
- Introdução

Introdução



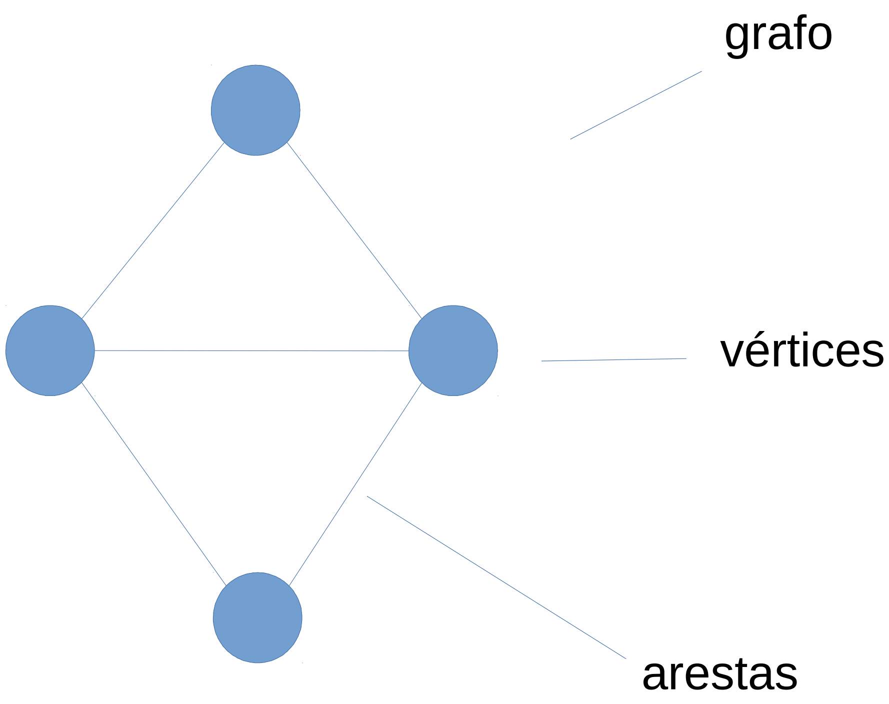


Introdução

- Grafos:
  - Um grafo **G** consiste de um conjunto finito de elementos chamado vértices (denotado por **V(G)**) e um conjunto de pares não ordenado de vértices chamado arestas (denotado por **E(G)**).

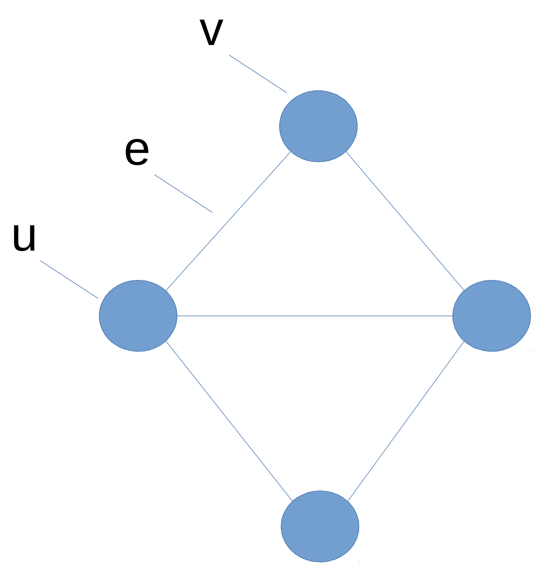
Introdução

- Grafos:



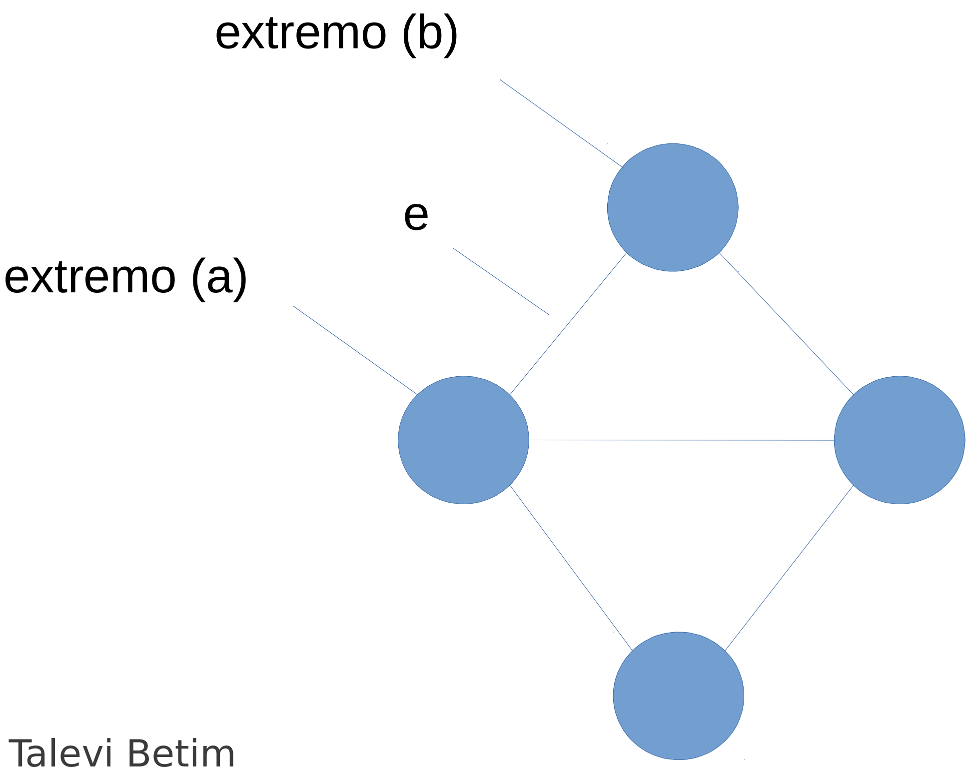
Introdução

- Grafos:
  - Se e é uma aresta **u** e **v** são seus vértices, podemos dizer que e liga **u** a **v**, e denotamos por **e = (u, v)** ou **e = uv.** Os vértices **u** e **v** são os extremos de **e**.



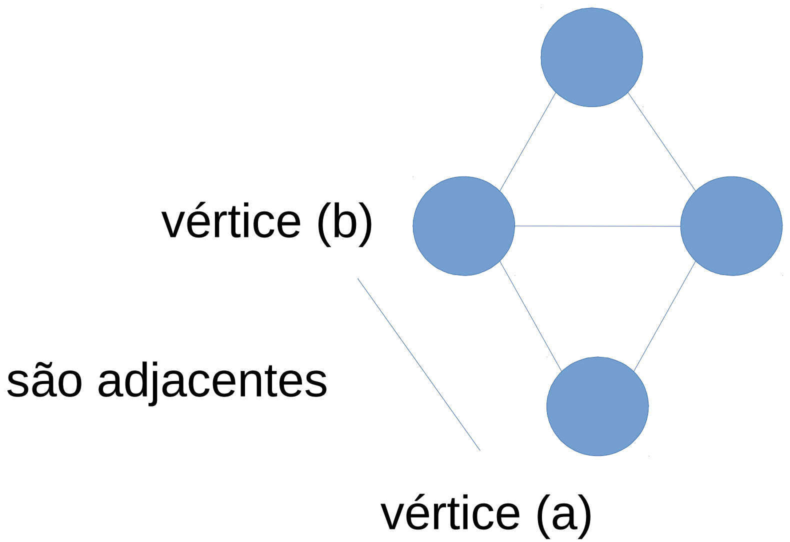
Introdução

- Grafos:
  - Os extremos de uma aresta são incidentes à aresta, e vice-versa.



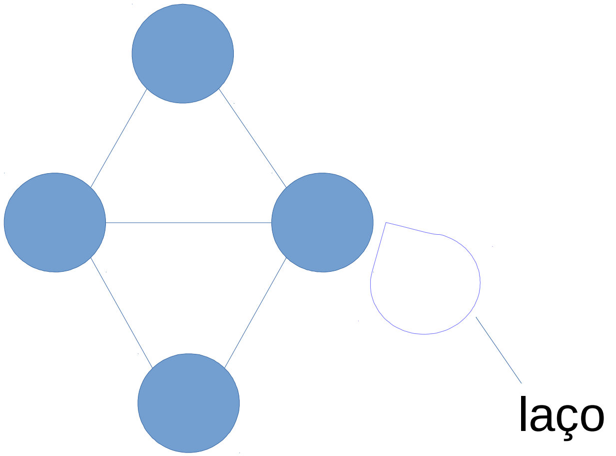
Introdução

- Grafos:
  - Dois vértices incidentes a uma mesma aresta são adjacentes. O mesmo nome é dado a duas arestas incidentes a um mesmo vértice (vizinhos).



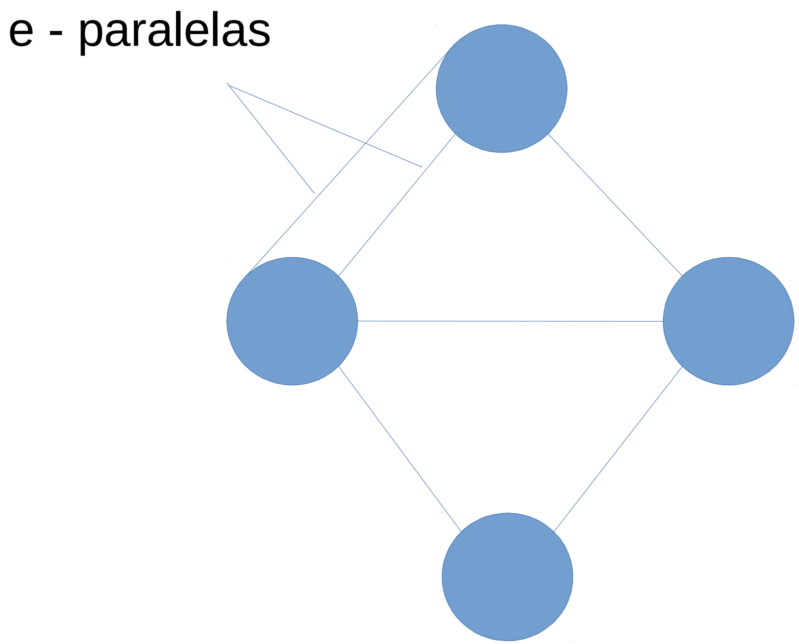
Introdução

- Grafos:
  - Laços:
    - Aresta com extremos no mesmo vértice.



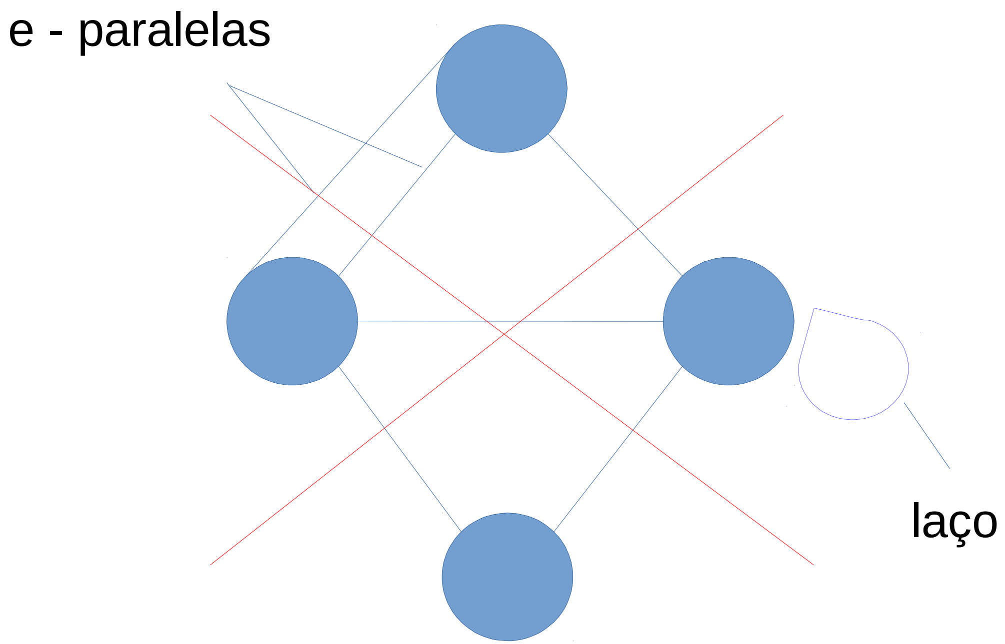
Introdução

- Grafos:
  - Arestas paralelas:
    - Ligam os mesmos pares de vértices



Introdução

- Grafos:
  - Grafo simples:
    - Não possui laço nem arestas paralelas

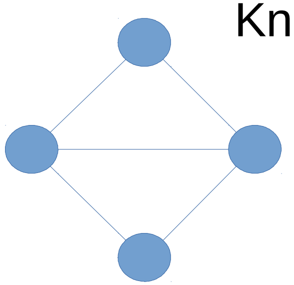


Introdução

- Grafos:
  - Ordem de um grafo:
    - Número de vértices que ele possui **v(G)** e **a(G)**:
      - Denotam, respectivamente, o número de vértices e arestas de um grafo **G**. Quando o grafo estiver subentendido usaremos apenas **v** e **a**.

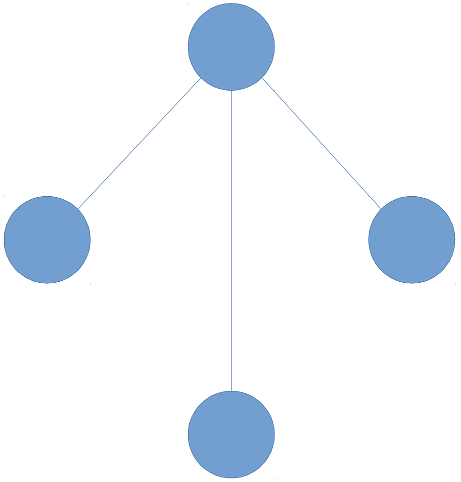
Introdução

- Grafos:
  - Um grafo é vazio se não possui vértices;
  - Um grafo é completo se cada par de vértices distintos é ligado por uma aresta;
  - Um grafo completo com **n** vértices é denotado por **kn**.



Introdução

- Grafos:
  - Um grafo é conexo se possui uma única componente.



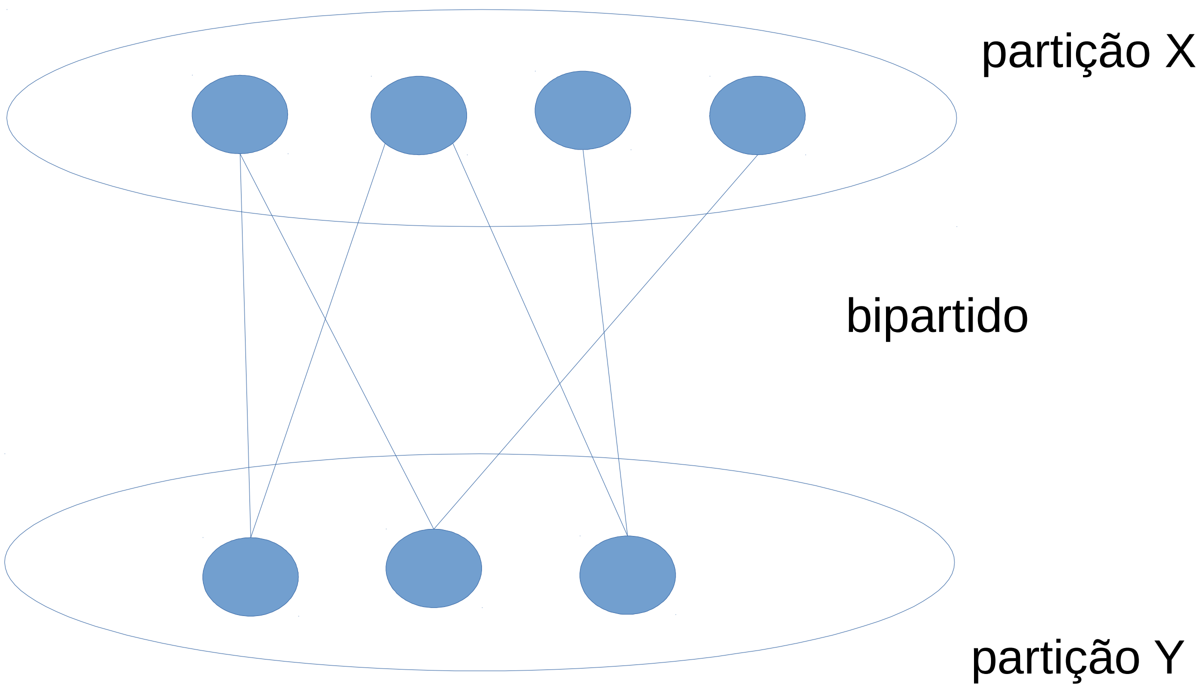
conexo

Introdução

- Grafos:
  - Um grafo é bipartido se o seu conjunto de vértices pode ser particionado em dois subconjuntos **X** e **Y**, tais que cada aresta possui um extremo em **X** e outro em **Y**; a partição **(X, Y)** é chamada de bipartição do grafo.
  - Um grafo é bipartido completo se ele é bipartido com bipartição **(X, Y)** e todo vértice de **X** é ligado a todo vértice de **Y**. Se |**X**| = **m** e |**Y| = n**, tal grafo é denotado por **Km, n**.

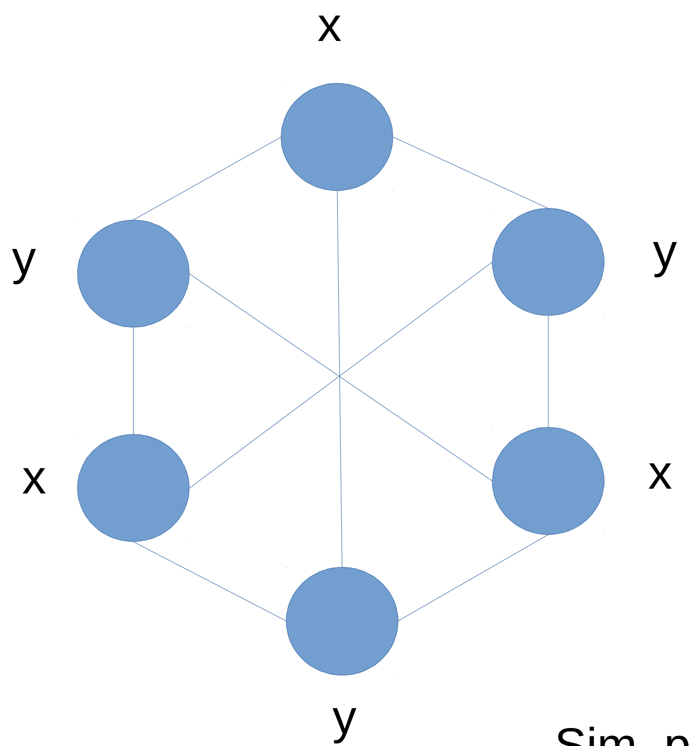
Introdução

- Grafos:



Introdução

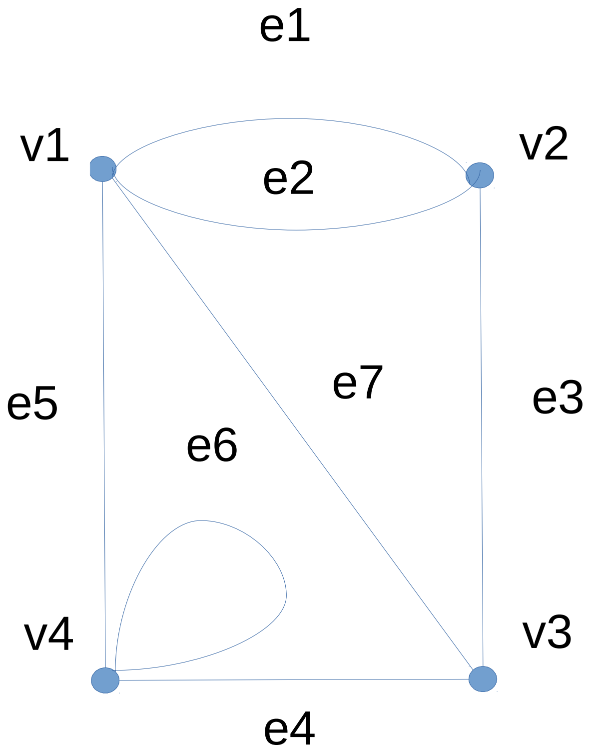
- Grafos:
  - É um grafo bipartido?



Sim, pois cada aresta liga x e y

Introdução

- Algoritmos em Grafos:
  - Como representar computacionalmente um grafo?

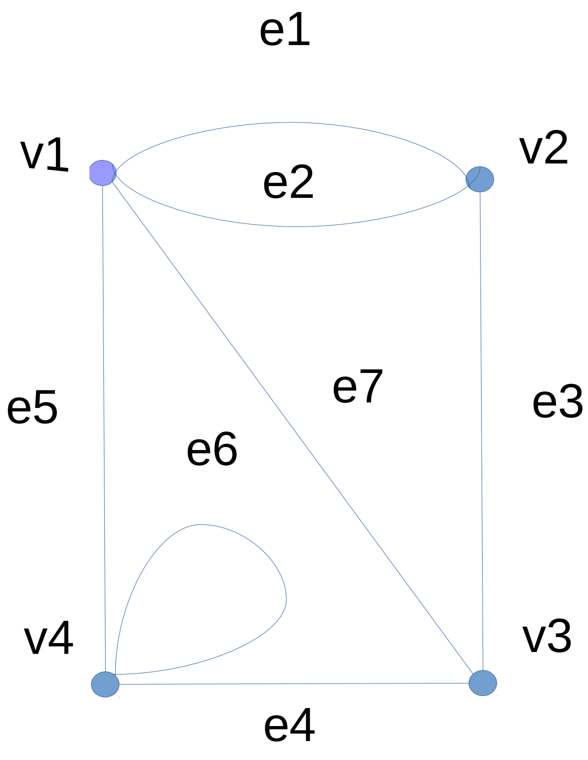


Matriz de incidência

G	e1	e2	e3	e4	e5	e6	e7
v1	1	1	0	0	1	0	1
v2	1	1	1	0	0	0	0
v3	0	0	1	1	0	0	1
v4	0	0	0	1	1	2	0

Introdução

- Algoritmos em Grafos:
  - Como representar computacionalmente um grafo?



Matriz de adjacência

G	v1	v2	v3	v4
v1	0	2	1	1
v2	2	0	1	0
v3	1	1	0	1
v4	1	0	1	1

Introdução

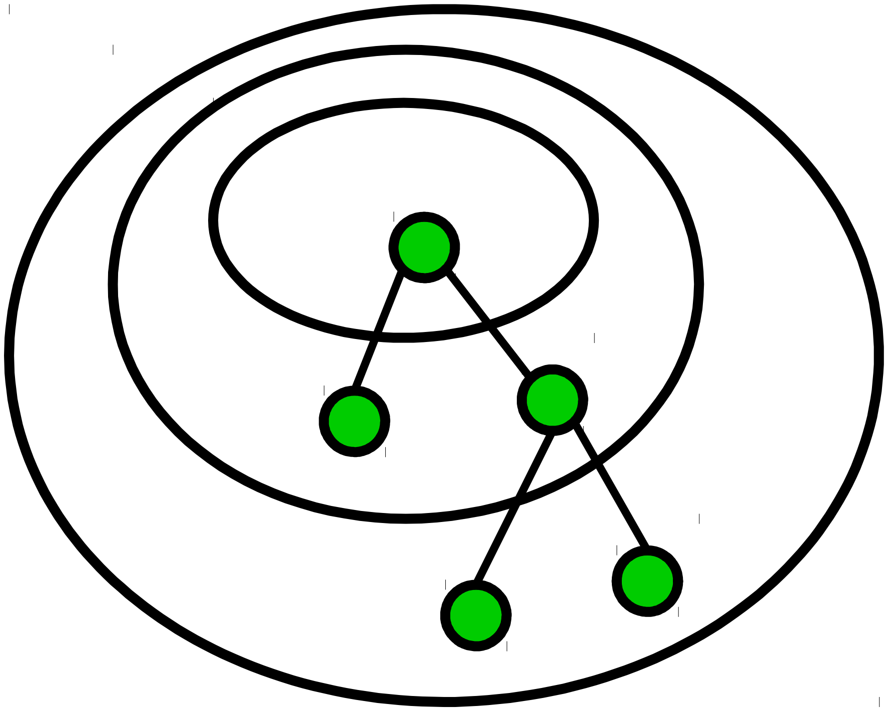
- Algoritmos em Grafos:
  - A estrutura de grafos possibilita a resolução de vários problemas, então é importante obter um processo sistemático e conscistente de como caminhar pelas arestas e vértices de uma grafo.
  - Vamos descrever como seria a busca em grafos nesta aula, a busca em largura e busca em profundidade.

Agenda

- Aula anterior
- Introdução
- Busca em largura

Busca em largura

- O algoritmo de busca em largura pode ser descrito da seguinte forma:



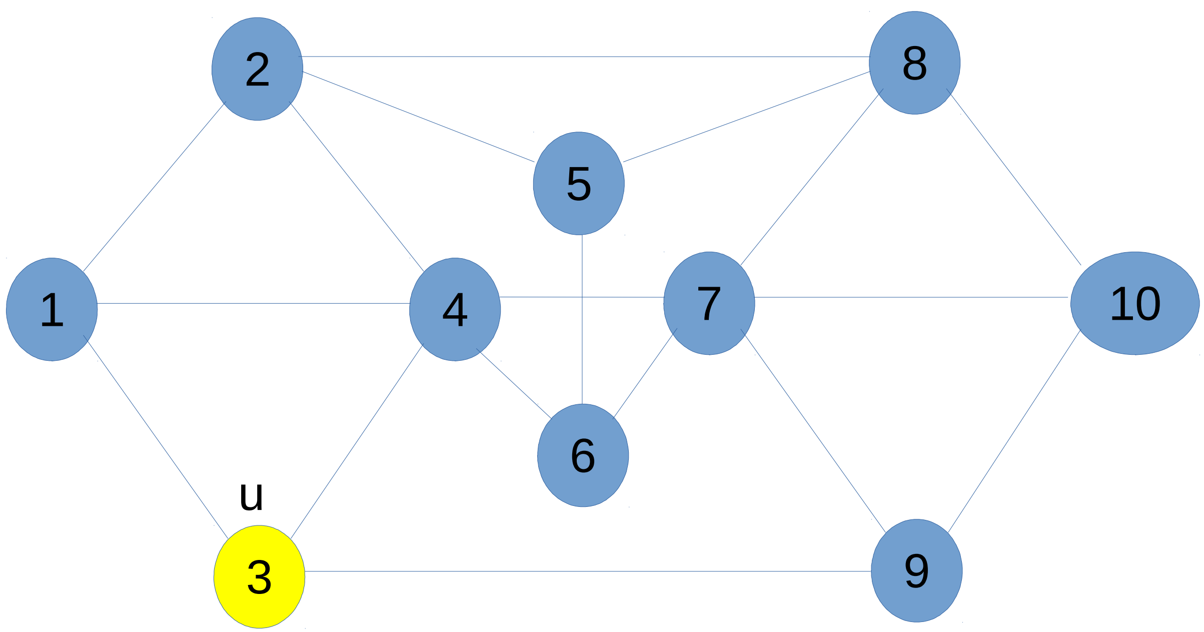
Busca em largura

- O algoritmo de busca em largura pode ser descrito da seguinte forma:
  - Põe na fila um vertice **u** qualquer de **G** e registra como alcançado;
  - Enquanto fila != de vazio faça
    - **u** ← elemento da frente da fila (retire **u** da fila)
    - Para toda aresta **(u, w)**, tal que **w** ainda não foi alcançado, marque **w** como alcançado e põe **w** na fila

Busca em largura

- O algoritmo de busca em largura pode ser descrito da seguinte forma:

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



Busca em largura

l1	escolha uma raiz s de G	c1	1
l2	marque s	c2	1
l3	insira s em F	c3	1
l4	enquanto F não está vazia faça	c4	n
l5	seja v o primeiro vértice de F	c5	n-1
l6	para cada w ∈ listaDeAdjacência de v faça	c6	$\sum_{j=1}^{n-1} t_j$
l7	se w não está marcado então	c7	$\sum_{j=1}^{n-1} t_j - 1$
l8	visite aresta entre v e w	c8	$\sum_{j=1}^{n-1} t_j - 1$
l9	marque w	c9	$\sum_{j=1}^{n-1} t_j - 1$
l10	insira w em F	c10	$\sum_{j=1}^{n-1} t_j - 1$
l11	senao se w ∈ F entao	c11	$\sum_{j=1}^{n-1} t_j - 1$
l12	visite aresta entre v e w	c12	$\sum_{j=1}^{n-1} t_j - 1$
l13	retira v de F	c13	n-1

Busca em largura

$$T(n)=C_1+C_2+C_3+C_4+C_5+C_6+C_7+C_8+C_9+C_{10}+C_{11}+C_{12}+C_{13}$$

$$T(n)=C_1+C_2+C_3+C_4n+C_5(n-1)+C_6(n-1)+C_7\sum_{j=1}^{n-1}t_j+C_8\sum_{j=1}^{n-1}t_j-1+C_9\sum_{j=1}^{n-1}t_j-1+C_{10}\sum_{j=1}^{n-1}t_j-1+C_{11}\sum_{j=1}^{n-1}t_j-1+C_{12}\sum_{j=1}^{n-1}t_j-1+C_{13}(n-1)$$

$$T(n)=C_1+C_2+C_3+C_4n+C_5(n-1)+C_6(n-1)+C_7\frac{n(n+1)}{2}-1+C_8\frac{n(n-1)}{2}+C_9\frac{n(n-1)}{2}+C_{10}\frac{n(n-1)}{2}+C_{11}\frac{n(n-1)}{2}+C_{12}\frac{n(n-1)}{2}+C_{13}(n-1)$$

$$T(n)=(\frac{c_7+c_8+c_9+c_{10}+c_{11}+c_{12}}{2})n^2+(c_4+c_5+c_6+c_{13}+\frac{c_7}{2}-\frac{c_8}{2}-\frac{c_9}{2}-\frac{c_{10}}{2}-\frac{c_{11}}{2}-\frac{c_{12}}{2})n-(c_5+c_6+c_7+c_{13})$$

Então, $T(n)=an^2+bn-c$ *em que a e b e c são constantes.* $T(n)$ *é uma função quadrática de n*

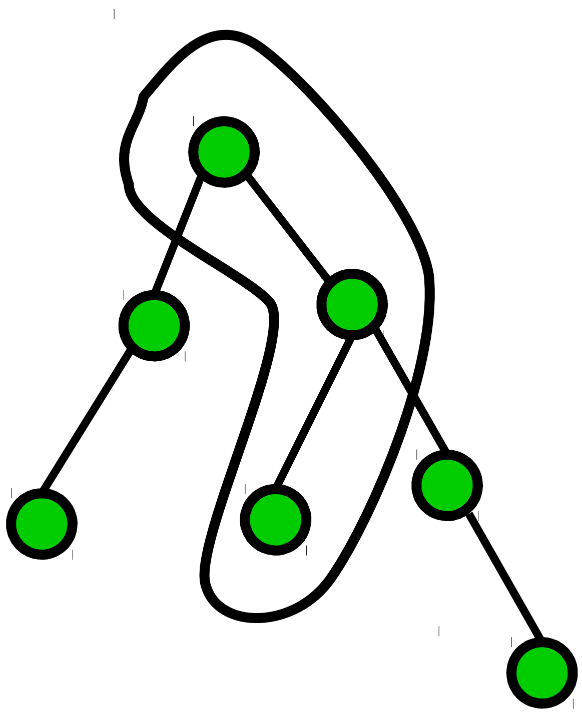
Mathias Talevi Betim

Agenda

- Aula anterior
- Introdução
- Busca em largura
- Busca em profundidade

Busca em profundidade

- O algoritmo de busca em profundidade pode ser descrito da seguinte forma:

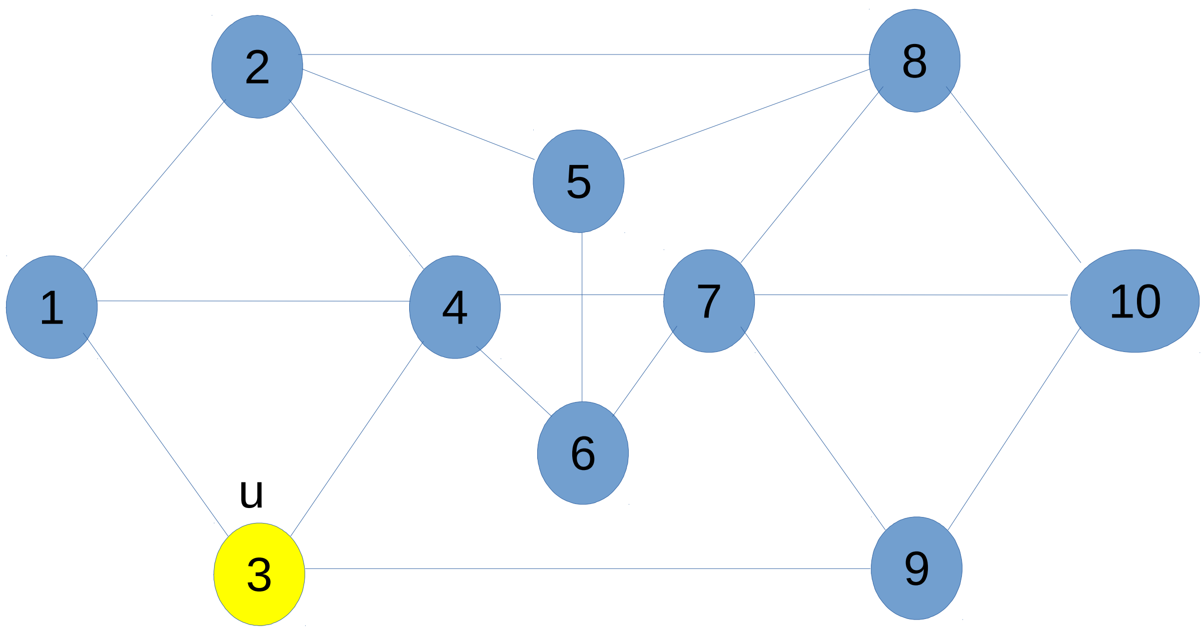


Busca em profundidade

- O algoritmo de busca em profundidade pode ser descrito da seguinte forma:
  - Empilhe um vértice qualquer **v** de **G** e marque-o como alcançado
  - Enquanto pilha != NULL faça
    - **v** ← elemento o topo da pilha
    - se existe aresta **(v,w)**, tal que **w** ainda não foi alcançado, então marque **w** como alcançado e empilhe **w**; senão desempilha **v**.

Busca em profundidade

- O algoritmo de busca em profundidade pode ser descrito da seguinte forma:



5
4
6
7
9
10
8
2
1
3

Busca em profundidade

l1	para u ← 1 até n faça	c1	n
l2	cor[u] ← branco	c2	n-1
l3	cor[r] ← cinza	c3	n-1
l4	p ← Cria-Pilha (r)	c4	n-1
l5	enquanto P não estiver vazia faça	c5	$\sum_{j=1}^{n-1} t_j$
l6	u ← Copia-Topo-da-Pilha (P)	c6	$\sum_{j=1}^{n-1} t_j - 1$
l7	v ← Próximo (Adj[u])	c7	$\sum_{j=1}^{n-1} t_j - 1$
l8	se v ≠ nil	c8	$\sum_{j=1}^{n-1} t_j - 1$
l9	então se cor[v] = branco	c9	$\sum_{j=1}^{n-1} t_j - 1$
l10	então cor[v] ← cinza	c10	$\sum_{j=1}^{n-1} t_j - 1$
l11	coloca-na-Pilha (v, P)	c11	$\sum_{j=1}^{n-1} t_j - 1$
l12	senão cor[u] ← preto	c12	$\sum_{j=1}^{n-1} t_j - 1$
l13	tira-da-Pilha (P)	c13	$\sum_{j=1}^{n-1} t_j - 1$
l14	devolva cor[1..n]	c14	n-1

Busca em profundidade

$$T(n)=C_1+C_2+C_3+C_4+C_5+C_6+C_7+C_8+C_9+C_{10}+C_{11}+C_{12}+C_{13}+C_{14}$$

$$T(n)=C_1 n+C_2(n-1)+C_3(n-1)+C_4(n-1)+C_5\sum_{j=1}^{n-1} t_j+C_6\sum_{j=1}^{n-1} t_j-1+C_7\sum_{j=1}^{n-1} t_j-1+C_8\sum_{j=1}^{n-1} t_j-1+C_9\sum_{j=1}^{n-1} t_j-1+C_{10}\sum_{j=1}^{n-1} t_j-1$$
$$+C_{11}\sum_{j=1}^{n-1} t_j-1+C_{12}\sum_{j=1}^{n-1} t_j-1+C_{13}\sum_{j=1}^{n-1} t_j-1+C_{14}(n-1)$$

$$T(n)=C_1 n+C_2(n-1)+C_3(n-1)+C_4(n-1)+C_5\frac{n(n+1)}{2}-1+C_6\frac{n(n-1)}{2}+C_7\frac{n(n-1)}{2}+C_8\frac{n(n-1)}{2}+C_9\frac{n(n-1)}{2}+C_{10}\frac{n(n-1)}{2}$$
$$+C_{11}\frac{n(n-1)}{2}+C_{12}\frac{n(n-1)}{2}+C_{13}\frac{n(n-1)}{2}+C_{14}(n-1)$$

$$T(n)=(\frac{c_5+c_6+c_7+c_8+c_9+c_{10}+c_{11}+c_{12}+c_{13}}{2})n^2+(c_1+c_2+c_3+c_4+c_{14}+\frac{c_5}{2}-\frac{c_6}{2}-\frac{c_7}{2}-\frac{c_8}{2}-\frac{c_9}{2}-\frac{c_{10}}{2}-\frac{c_{11}}{2}-\frac{c_{12}}{2}-\frac{c_{13}}{2})n-(c_2+c_3+c_4+c_5+c_{14})$$

Então, $T(n)=an^2+bn-c$ *em que a e b e c são constantes.* $T(n)$ *é uma função quadrática de n*

Mathias Talevi Betim

Agenda

- Aula anterior
- Introdução
- Busca em largura
- Busca em profundidade
- Exercícios

Exercícios

- Implemente o grafo solicitado em sala (desenho no quadro) e codifique a busca em largura com fila e profundidade com pilha.
- Analise a complexidade dos algoritmos de busca

Agenda

- Aula anterior
- Introdução
- Busca em largura
- Busca em profundidade
- Exercícios
- Próxima aula

Próxima aula

- Método guloso

AULA 15

Prof. Mathias