

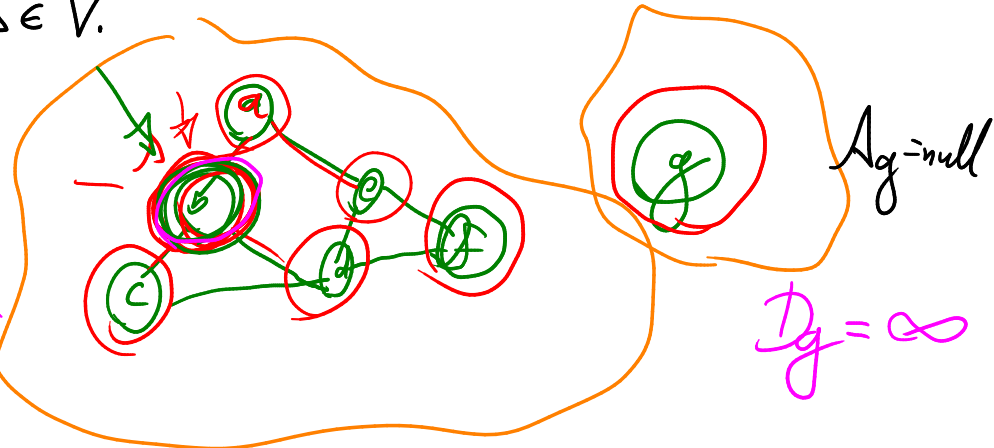
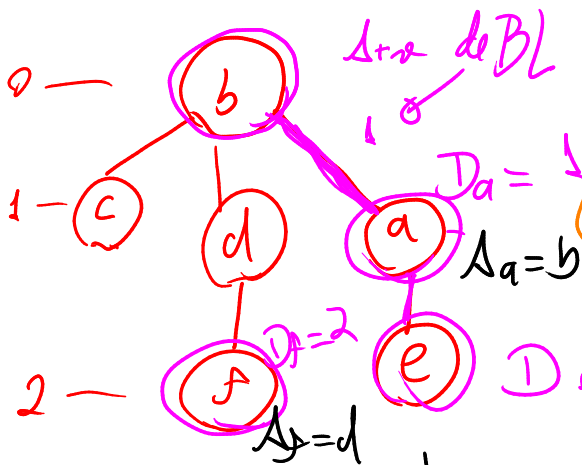
3. Buscas em Grafos (pg 25)

3.1. Busca em Largura

→ Entrada: • um grafo $G=(V,E)$;
 • um vértice de origem $s \in V$.

→ dirigido
→ não-dirigido
→ arcos ou arestas

→ Por nível



→ Encontra a distância mínima em número de arestas/arcos de um vértice s para todo vértice $v \in V$.

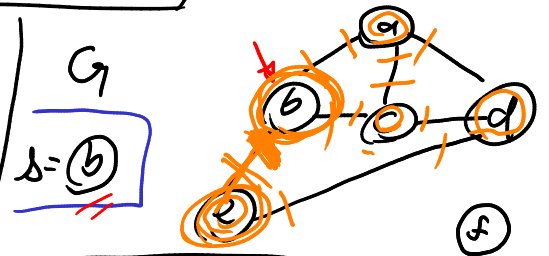
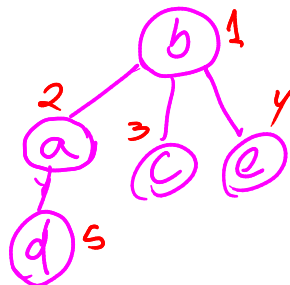
Teste de Mesa: Busca em Largura:

Algoritmo 3: Busca em largura.

Input : um grafo $G=(V,E)$, vértice de origem $s \in V$
 // configurando todos os vértices
 1 $C_v \leftarrow \text{false} \forall v \in V$
 2 $D_v \leftarrow \infty \forall v \in V$
 3 $A_v \leftarrow \text{null} \forall v \in V$
 // configurando o vértice de origem
 4 $C_s \leftarrow \text{true}$
 5 $D_s \leftarrow 0$
 // preparando fila de visitas
 6 $Q \leftarrow \text{Fila}()$
 7 $Q.\text{enqueue}(s)$
 // propagação das visitas
 8 **while** $Q.\text{empty}() = \text{false}$ **do**
 9 $u \leftarrow Q.\text{dequeue}()$
 10 **foreach** $v \in N(u)$ **do**
 11 **if** $C_v = \text{false}$ **then**
 12 $C_v \leftarrow \text{true}$
 13 $D_v \leftarrow D_u + 1$
 14 $A_v \leftarrow u$
 15 $Q.\text{enqueue}(v)$
 16 **return** (D, A)

Fila $Q = (\text{b}, \text{c}, \text{d}, \text{a}, \text{f}, \text{e})$

Arv. de Busca em Largura



	C	D	A
a	T	1	b
b	T	0	"
c	T	1	b
d	T	2	a
e	T	1	b
f	f	"	"

3.2 Busca em Profundidade

→ Entrada: · um grafo $G = (V, E)$;
· um vértice de origem $s \in V$.

→ Não garante a distância mínima ou número de arestas/arços;

→ Não faz visita por níveis;

→ Pilha ou Recursividade.

