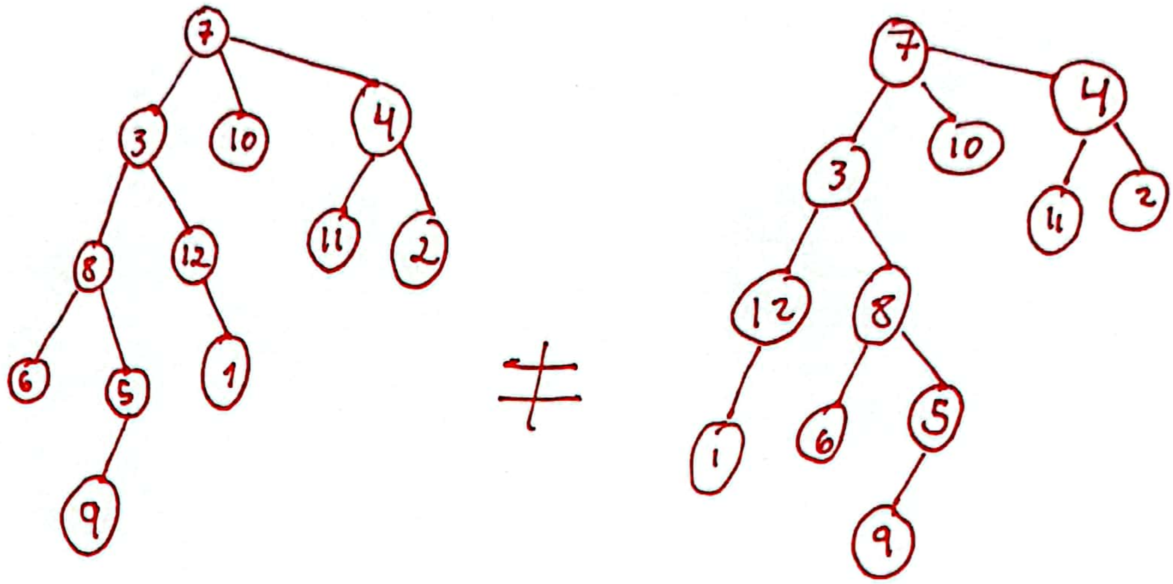
Árvore Ordenada

Árvore enraizada em que os filhos são ordenados

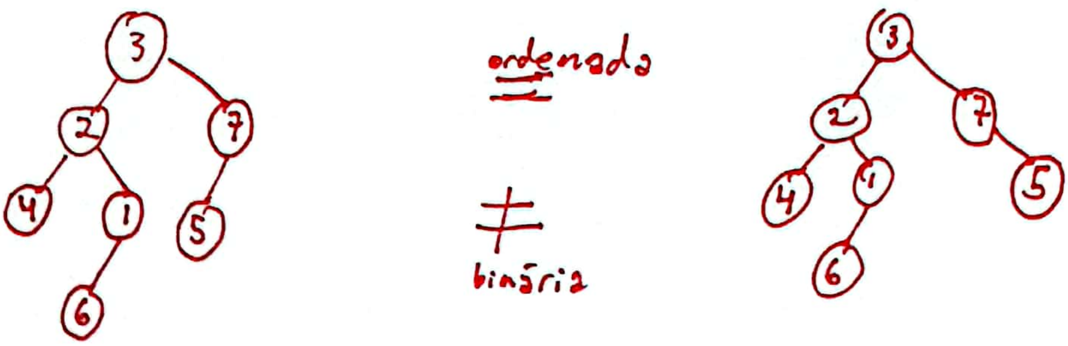


Árvore Binária

Não contém nós

Três conjuntos disjuntos de vértices

- Nó raiz
- Árvore binária (subárvore esquerda)
- Árvore binária (subárvore direita)



Capítulo 22

Buscas

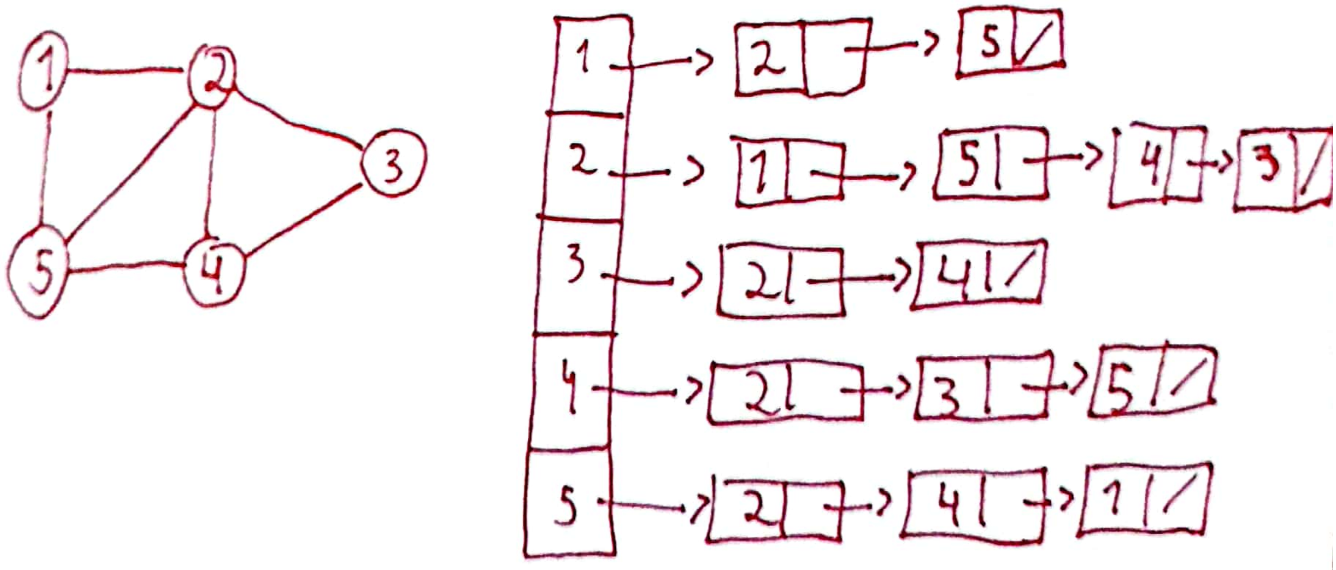
Sistematicamente "visitam/descobrem" os vértices "caminhando" pelas arestas

Representação de Grafos
Listas de adjacências
Matrizes de adjacências

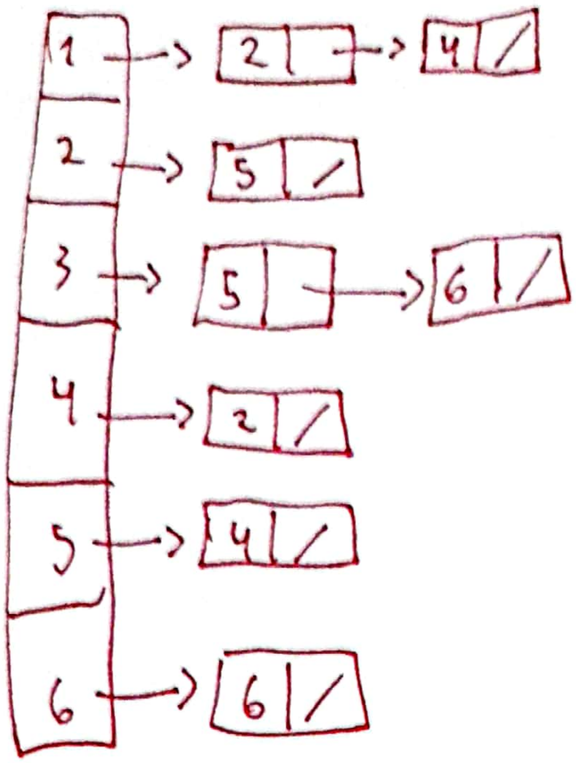
Listas de adjacências

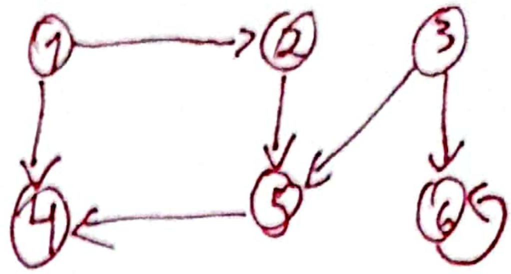
$G = (V, E)$

Adj vetor $|V|$ listas
- uma lista para cada vértice



MC558





Matrizes de adjacencias

$$A = (a_{ij})$$

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

	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

MC558

11/04

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

Busca em largura (B F S)

raiz s (source) — Breadht first search

Dados: $G = (V, E)$

$s \in V$

Técnica

vertices são coloridos

branco: não descoberto

cinza: descoberto, não visitado

preto: visitado

11/04

MC 558

BFS (G, s)

```
FOR each u ∈ V \ {s}
    DO color(u) ← BRANCO
        d(u) ← ∞
        π(u) ← NIL
Color(s) ← cinza
d(s) ← 0
π(s) ← NIL
Q ← ∅
ENQUEUE (Q, s) // enfileirar
while Q ≠ ∅
    DO u ← DEQUEUE (Q) // desenfileirar
        FOR each v ∈ Adj(u)
            if color(v) = BRANCO
                Then color(v) ← cinzA
                    d(v) ← d(u) + 1
                    π(v) ← u
                    ENQUEUE(Q, v)
        Color(u) ← PRETO
```