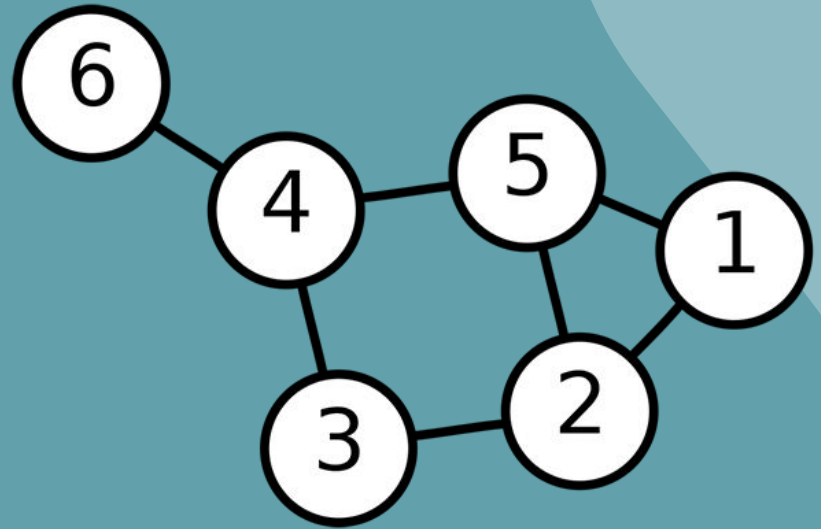
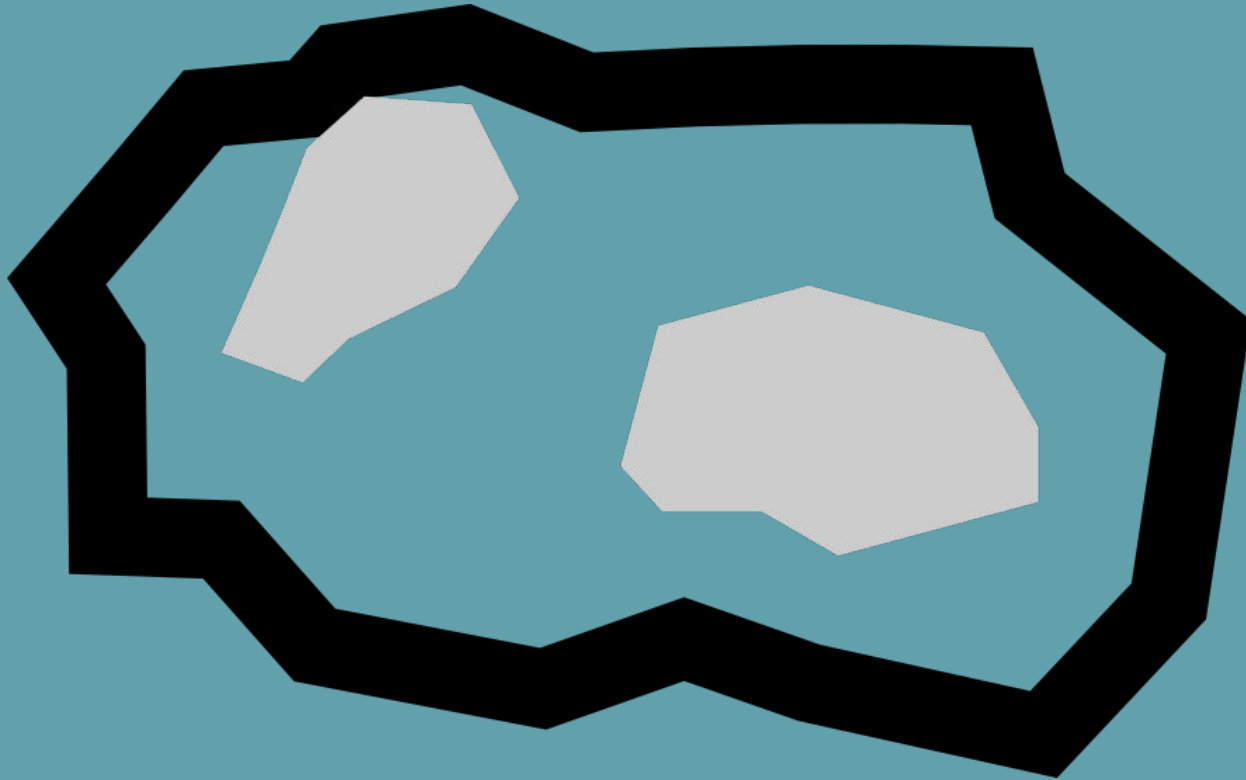


GRAFOS:

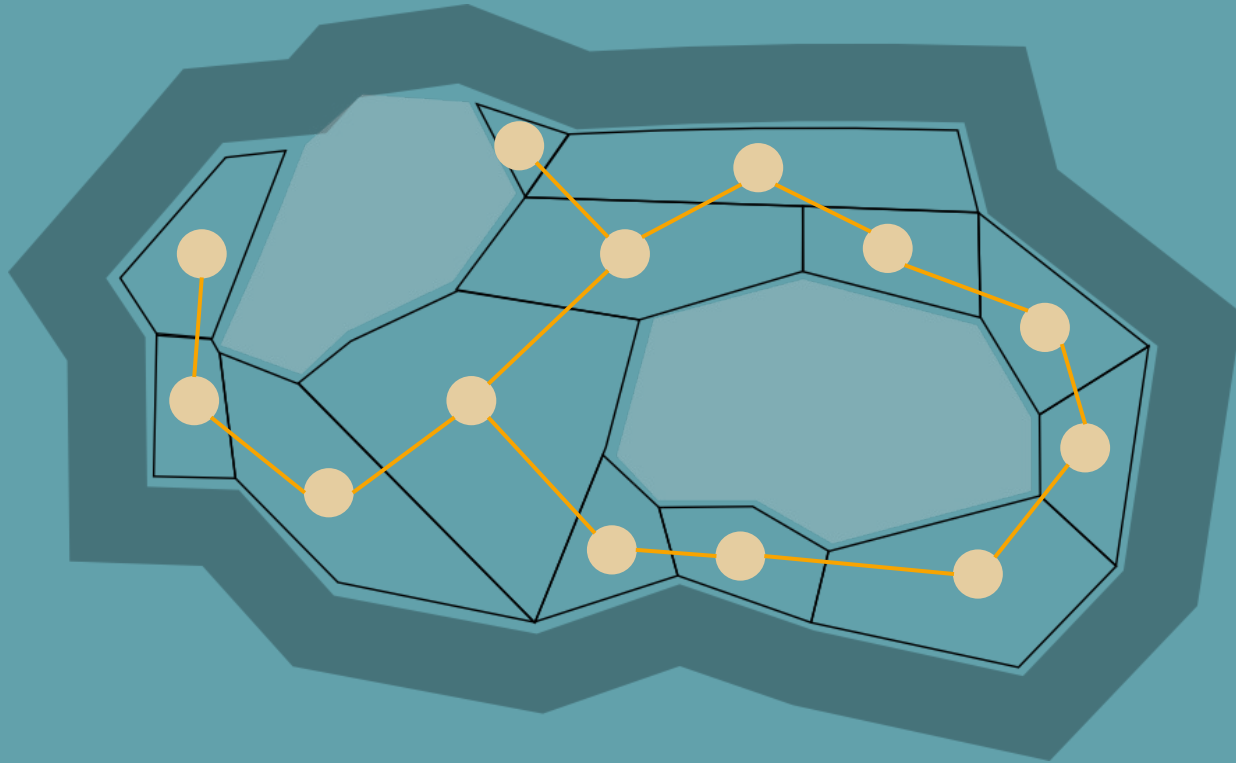
Busca em Profundidade



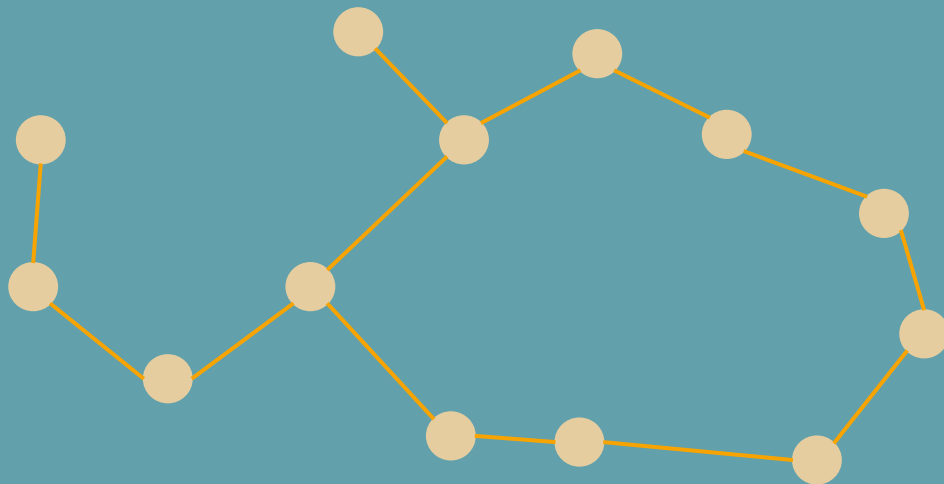
Busca em Grafos



Busca em Grafos



Busca em Grafos



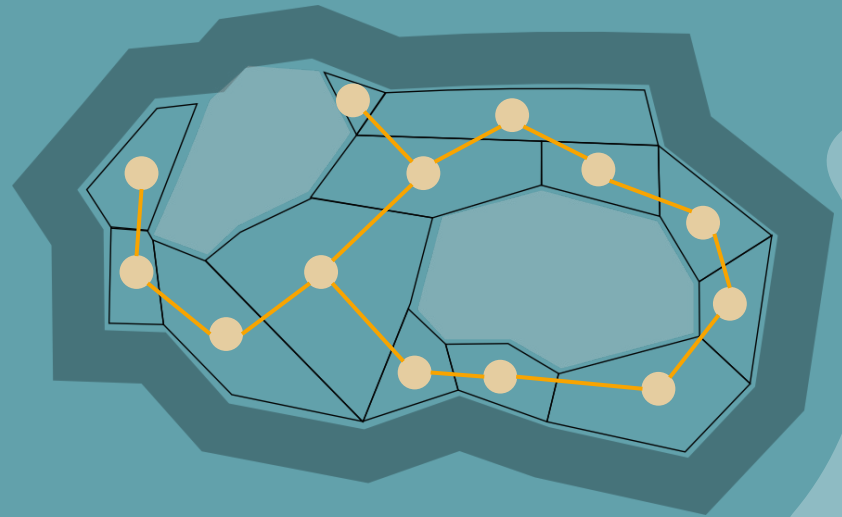
Busca em Grafos

Buscas, varredura ou pesquisa em grafos são estratégias algorítmicas usadas para percorrer os vértices de um grafo

- Seguem/respeitam as arestas

Duas estratégias de pesquisa

1. Busca em largura
(*Breadth-First Search* - BFS)
2. Busca em profundidade
(*Depth-First Search* - DFS)

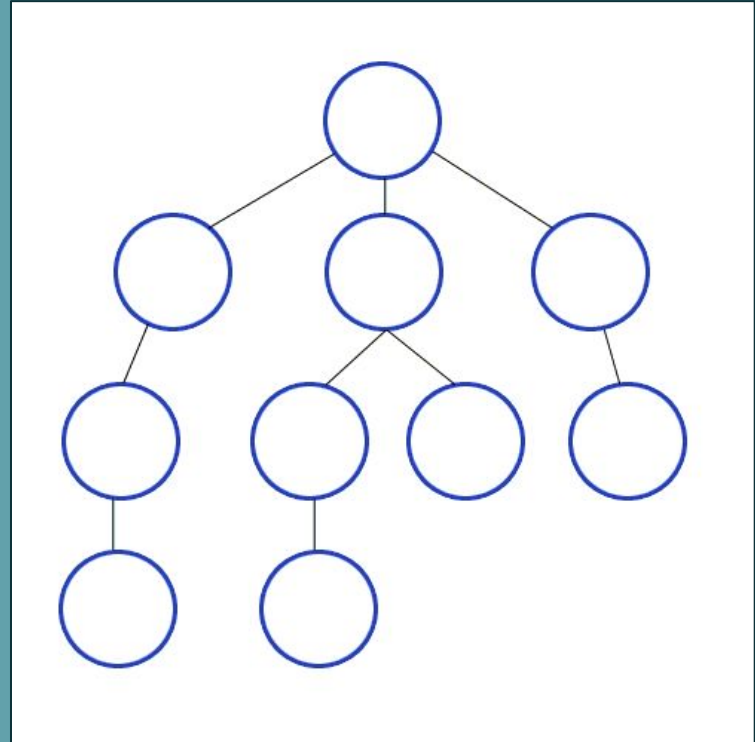


Busca em Profundidade

(Depth-First Search - DFS)

Busca em Profundidade (depth-first search - DFS)

- Tenta seguir sempre o mais “fundo” possível



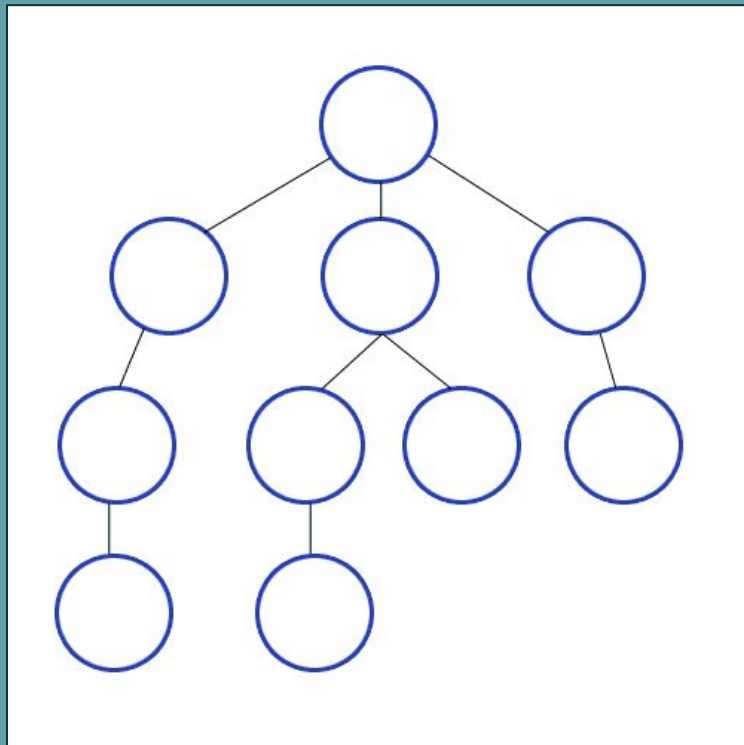
Busca em Profundidade

(depth-first search - DFS)

Cada vértice pode estar em três estados

- **Branco**: não-visitado
- **Cinza**: pesquisa iniciada
- **Preto**: pesquisa terminada

O vértice se torna **Cinza** logo que a pesquisa o atinge, e **Preto** quando todos os seus vizinhos já foram analisados



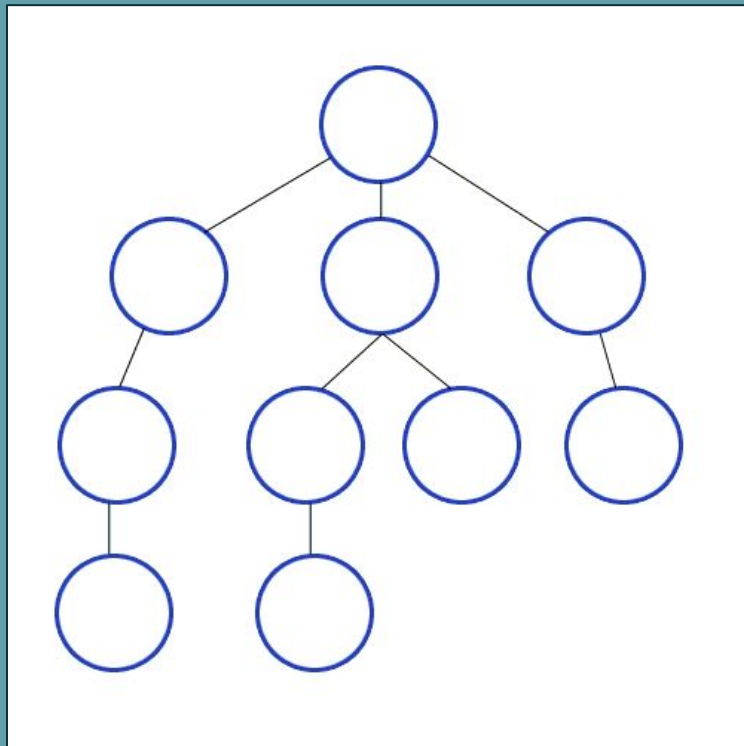
Busca em Profundidade

(depth-first search - DFS)

Vamos assumir que a pesquisa usa as seguintes variáveis

- **cor[u]** - conforme explicado anteriormente
- **i[u]** - momento em que a visita a u é iniciada
- **f[u]** - momento em que a visita a u é finalizada
- **ante[u]** - vértice antecessor de u na pesquisa

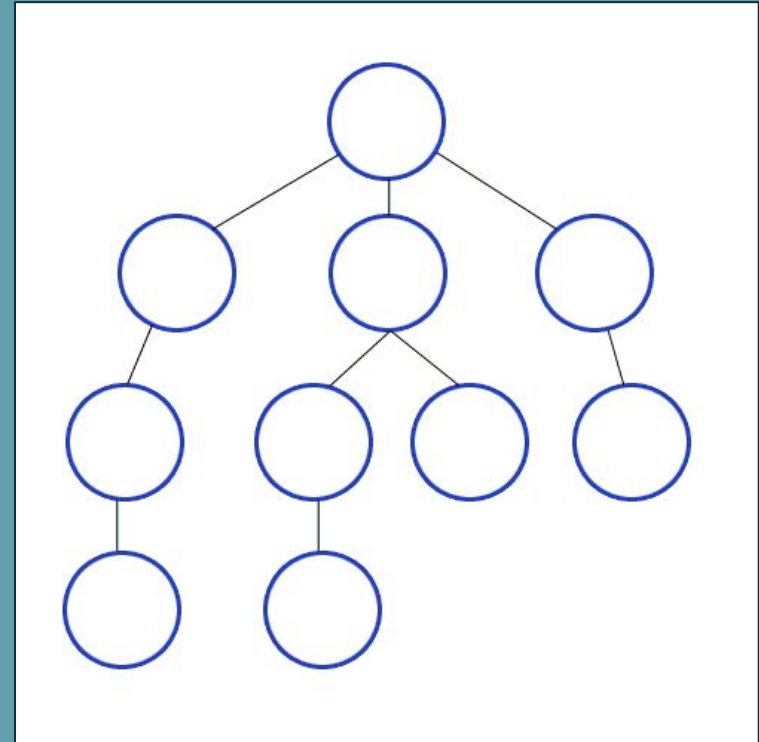
Para setar **i[u]** e **f[u]**, vamos usar uma variável global tempo



Busca em Profundidade

(depth-first search - DFS)

- Vamos assumir que a busca em profundidade tem um vértice de início s
- Antes de iniciar a busca propriamente dita, é preciso inicializar as variáveis
- Vamos definir um método DFS-START() para o início da busca



Busca em Profundidade

(depth-first search - DFS)

Inicialização

DFS-START(grafo G , vértice s)

para todo vértice v

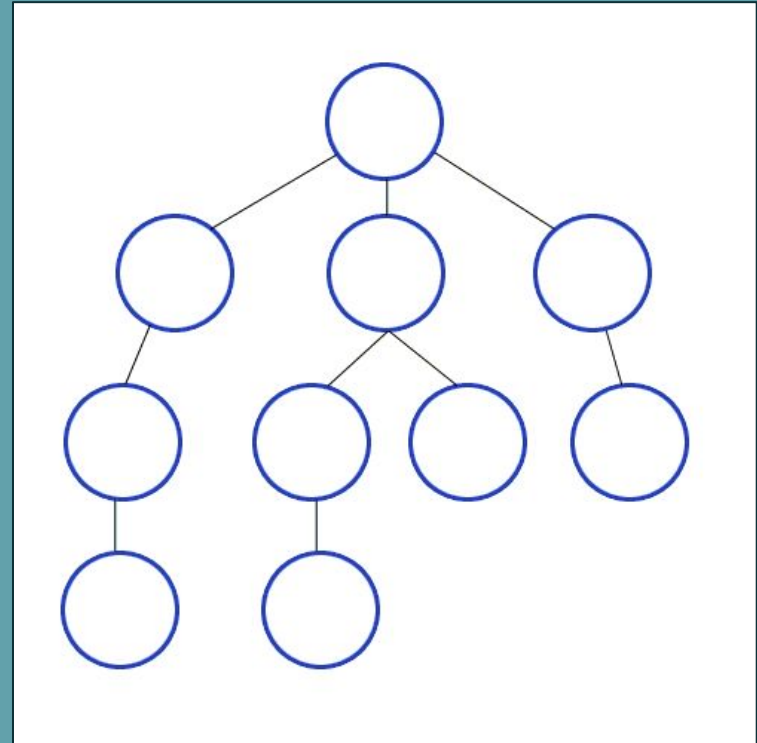
cor[v] = BRANCO

i[v] = **f**[v] = -2

ante[v] = -2

tempo = 1

DFS-VISIT(G, s)



Busca em Profundidade

(depth-first search - DFS)

DFS-VISIT(grafo G , vértice u)

cor[u] = Cinza;

i[u] = tempo ++;

para cada v de Adj[u]

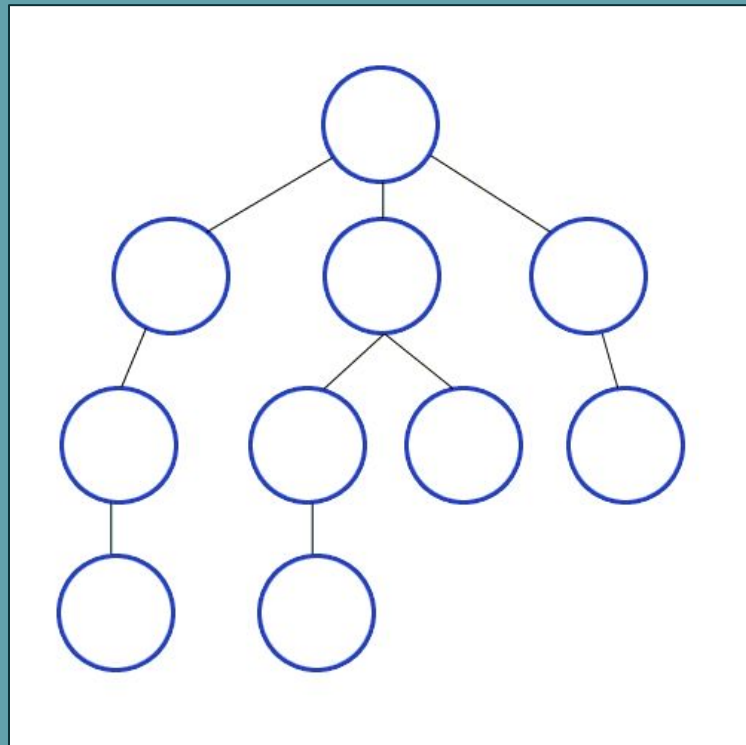
if **cor**[v] = Branco

ante[v] = u ;

DFS-VISIT(G, v);

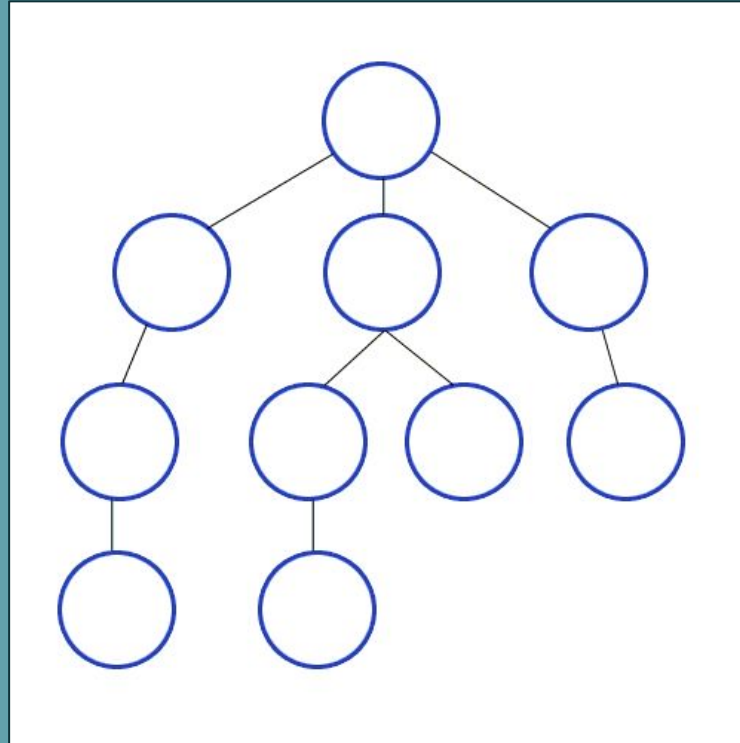
cor[u] = Preto;

f[u] = tempo ++;



Busca em Profundidade

(depth-first search - DFS)



Busca em Profundidade (depth-first search - DFS)

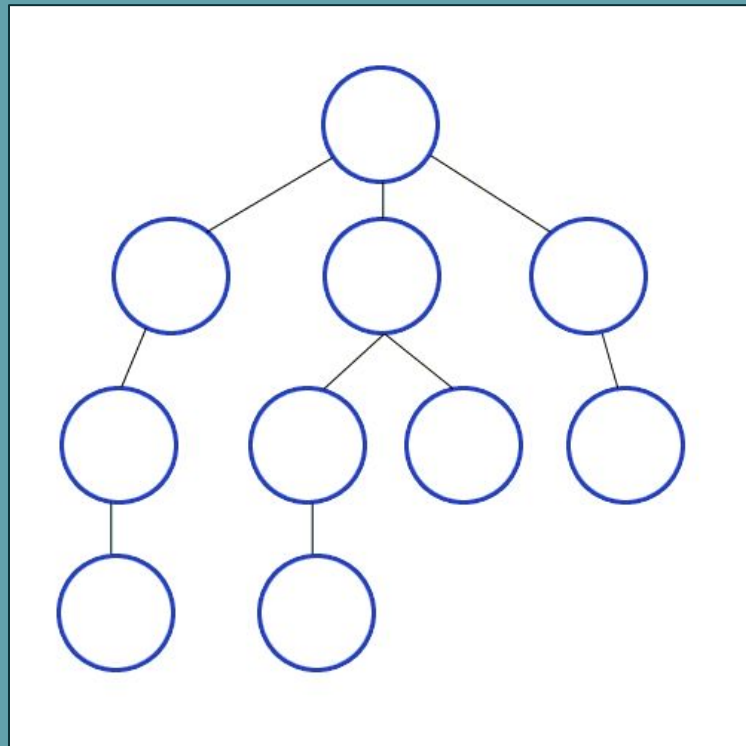
(depth-first search - DFS)

As principais saídas do algoritmo são os arrays:

- ante[]
- i[] e f[]

O array **ante[]** também guarda caminhos:

- Cada posição **ante[v]** guarda o antecessor de um vértice **v** no caminho encontrado



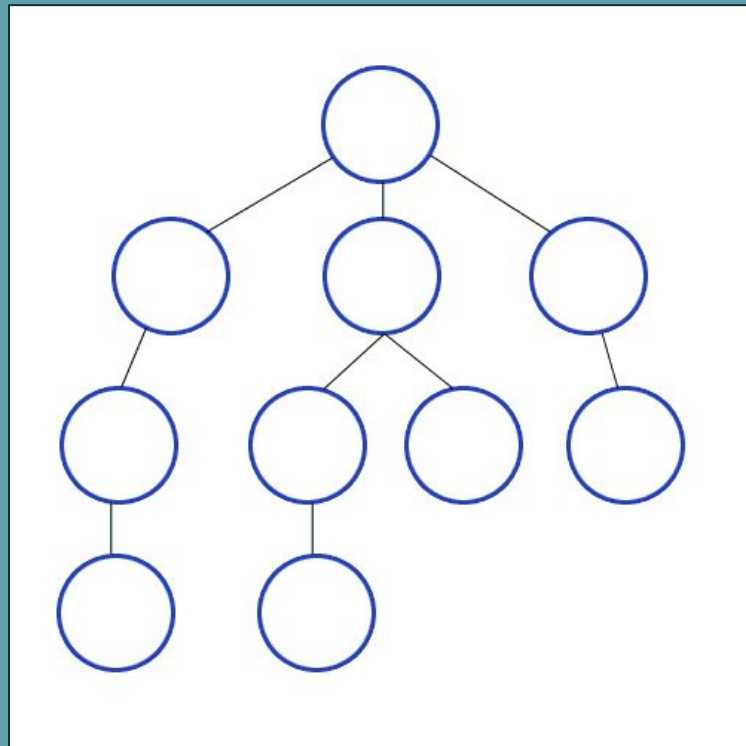
Busca em Profundidade

(depth-first search - DFS)

A Busca em Profundidade serve para achar caminhos **quaisquer** de um vértice de origem *s* a cada um dos outros vértices.

Os caminhos encontrados pela Busca em Profundidade podem ser todos representados como uma árvore.

Essa estratégia pode ser usada como base para algoritmos que resolvem outros problemas.



OBRIGADO!

DÚVIDAS?

marcos.azevedo@unicap.br

CREDITS: This presentation template was created by **Slidesgo**,
including icons by **Flaticon** and infographics & images by
Freepik