Projeto e Análise de Algoritmos
2024.2

DFS e Aplicações

Prof. Marcio Costa Santos DCC/UFMG

Busca em Grafos

- Vistar os vértices do grafo em ordem.
- Necessidade em muitos algoritmos.
- Principal operação de uma estrutura de dados.
- Propriedades distintas.
- Aplicável a grafos e dígrafos.

Busca em Profundidade

- Vamos visitar os vértices *filhos* de um nó antes dele.
- Vamos manter algumas variáveis:
  - $\pi[v]$: pai do vértice v.
  - $i[v]$: tempo que encontrados o vértice v.
  - $f[v]$: tempo que visitamos o vértice v.
- vértices brancos, cinzas e pretos.

Busca em Profundidade - Algoritmo

Entrada: Grafo $G = (V, s)$, vértice inicial s.
para v *até* $V(G)$ **faça**
    $color[v] \leftarrow BRANCO$;
    $\pi[u] = \lambda$;
fim
$time \leftarrow 0$;
para $v \in V(G)$ **faça**
    se $color[v] == BRANCO$ **então**
        `DFS-VISIT`(G, u);
    fim
fim

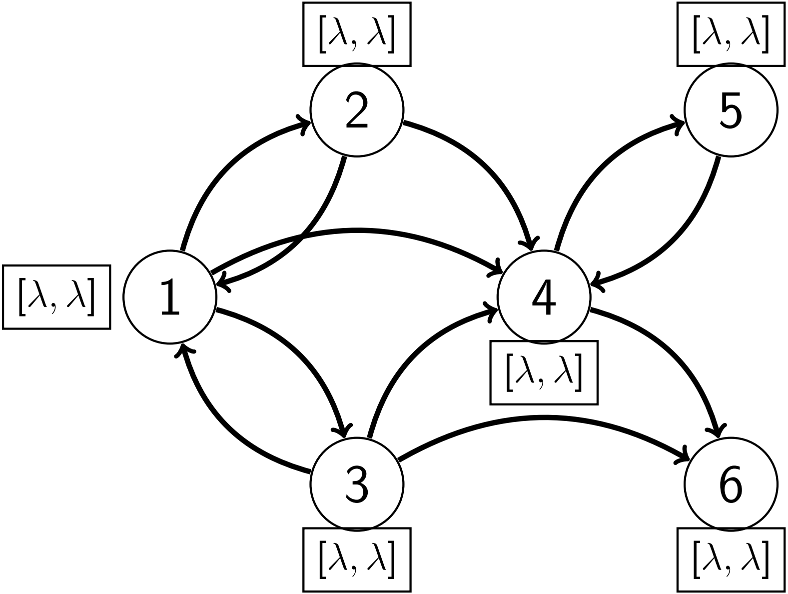
Algoritmo 1: DFS(G,s)

Busca em Profundidade - Algoritmo

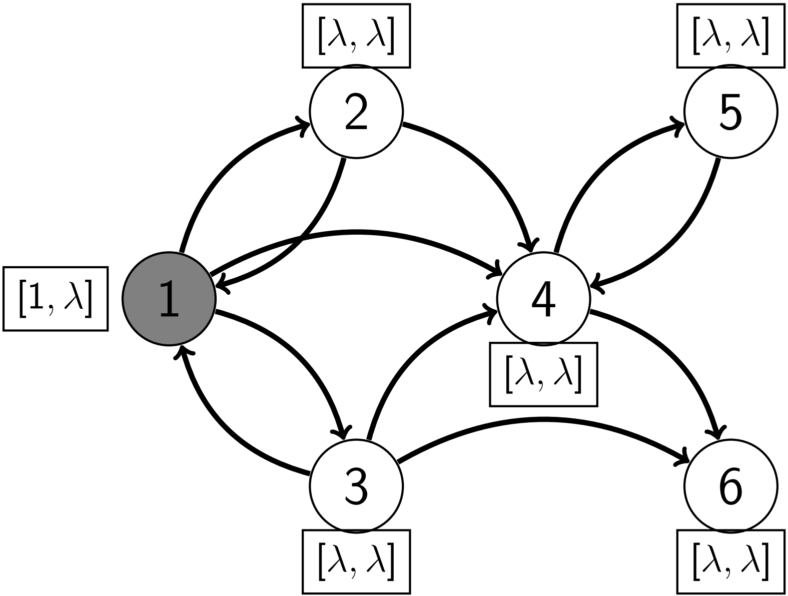
$time \leftarrow time + 1;$
$color[v] = CINZA; \ i[v] = time;$
para $u \in N(v)$ **faça**
  se $color[u] = BRANCO$ **então**
    $\pi[u] = v;$
    DFS-VISIT(G,u);
  fim
fim
$time \leftarrow time + 1;$
$color[v] = PRETO; \ f[v] = time;$

Algoritmo 2: DFS-VISIT(G,v)

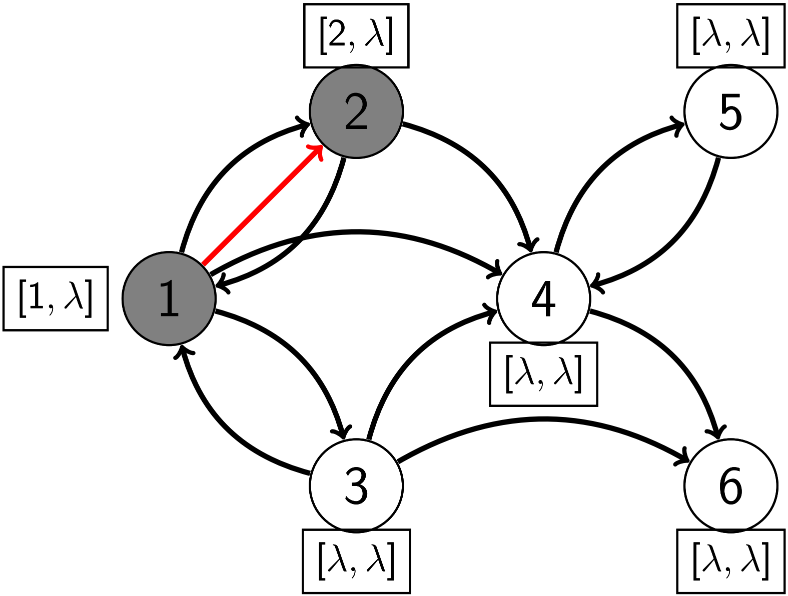
Busca em Profundidade - Exemplo



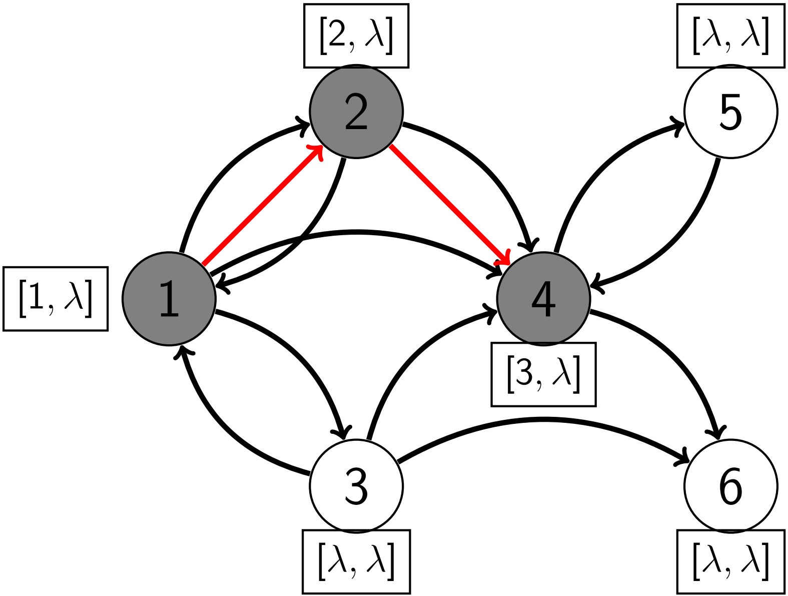
Busca em Profundidade - Exemplo



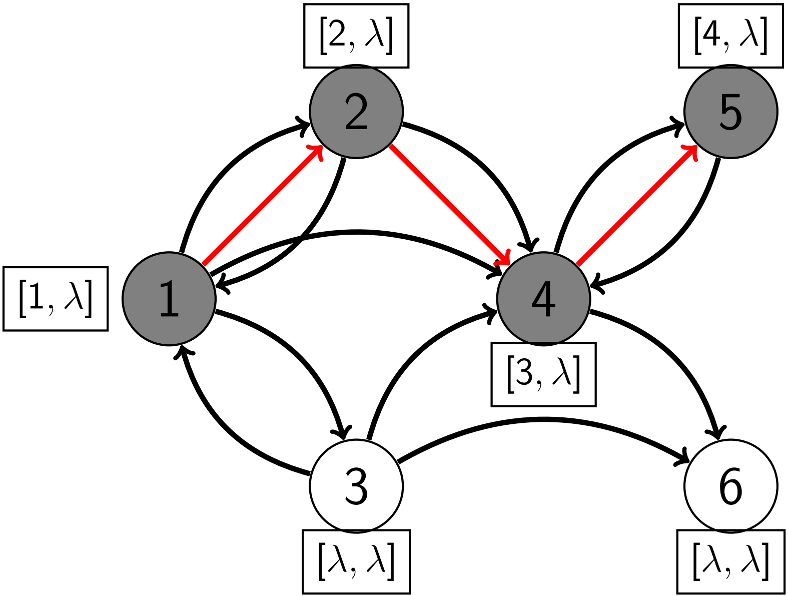
Busca em Profundidade - Exemplo



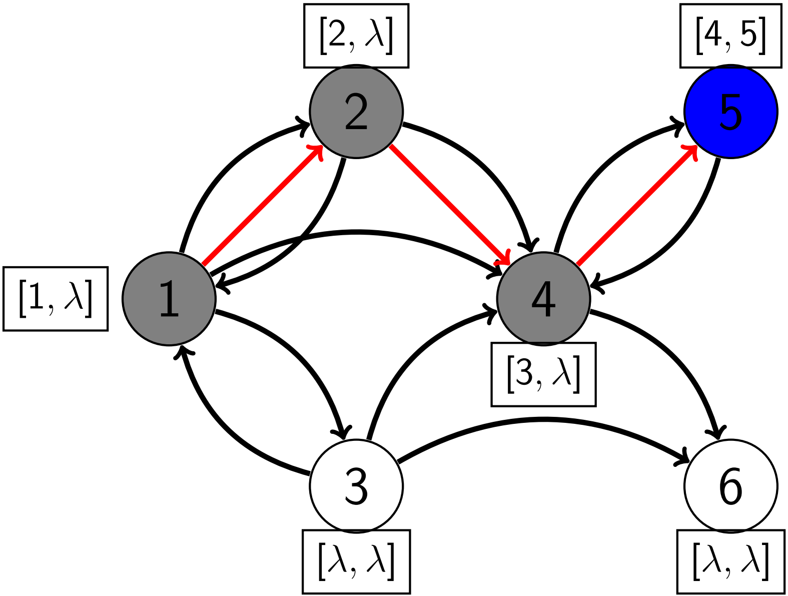
Busca em Profundidade - Exemplo



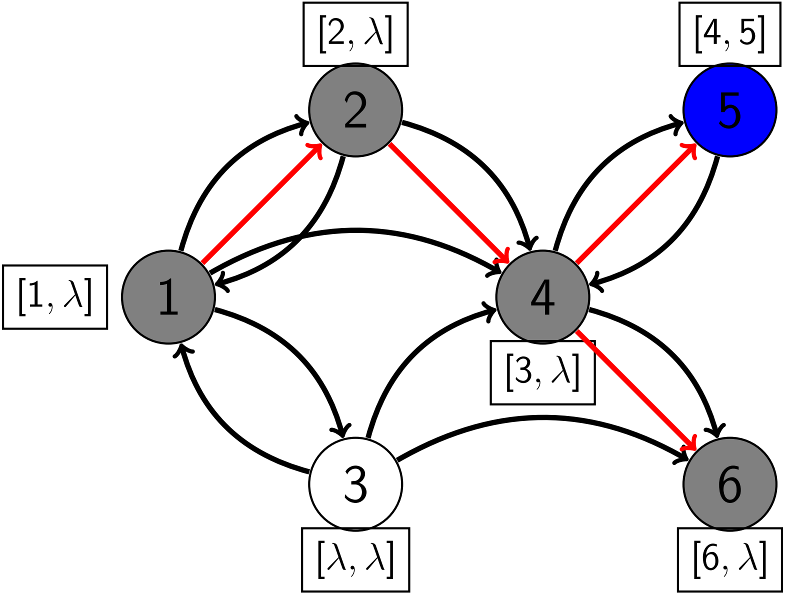
Busca em Profundidade - Exemplo



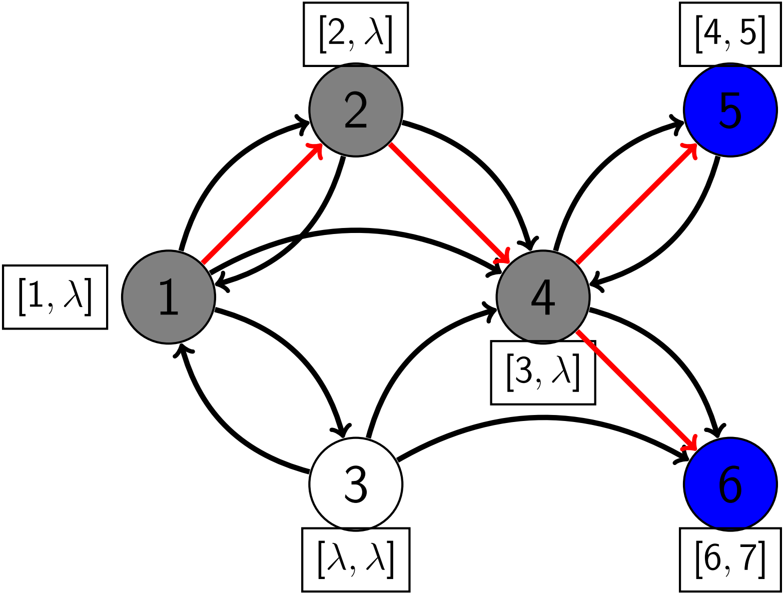
Busca em Profundidade - Exemplo



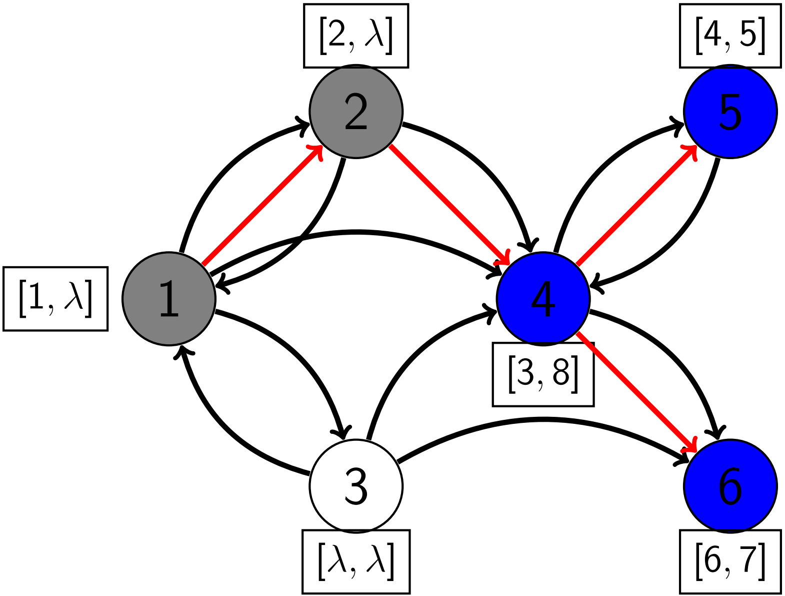
Busca em Profundidade - Exemplo



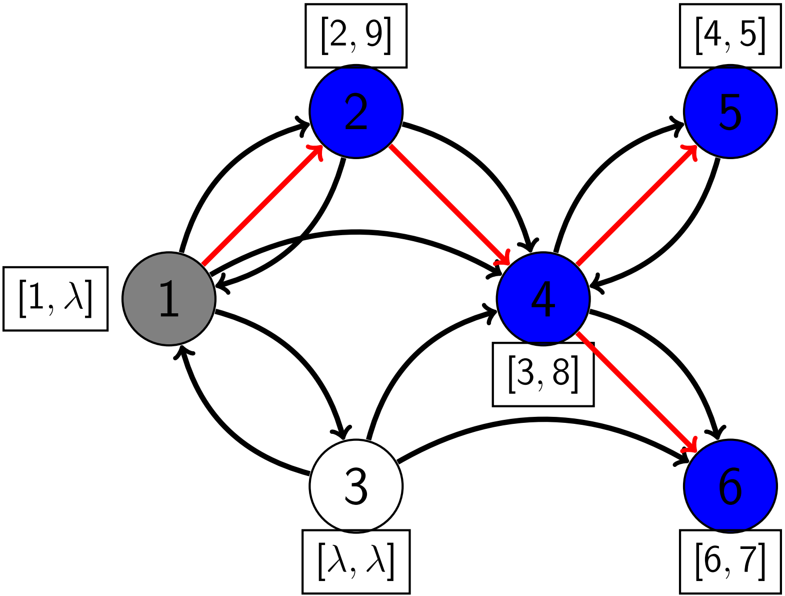
Busca em Profundidade - Exemplo



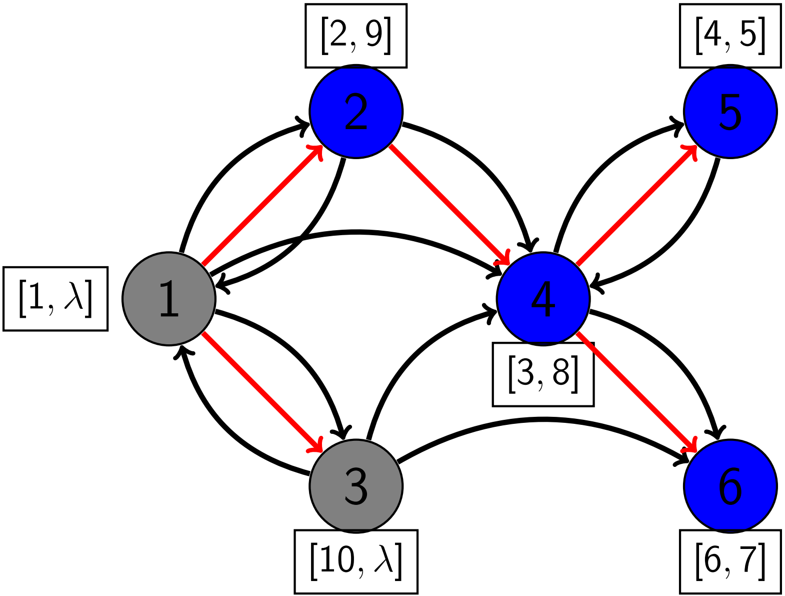
Busca em Profundidade - Exemplo



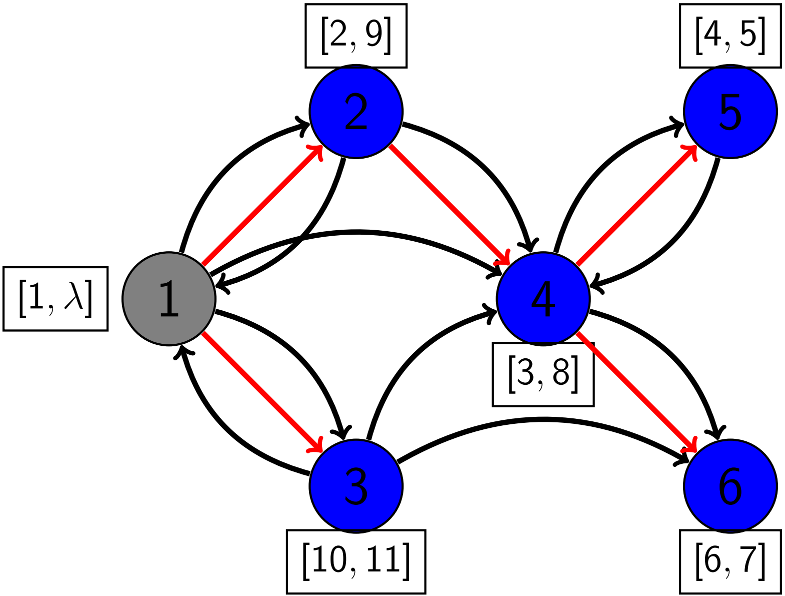
Busca em Profundidade - Exemplo



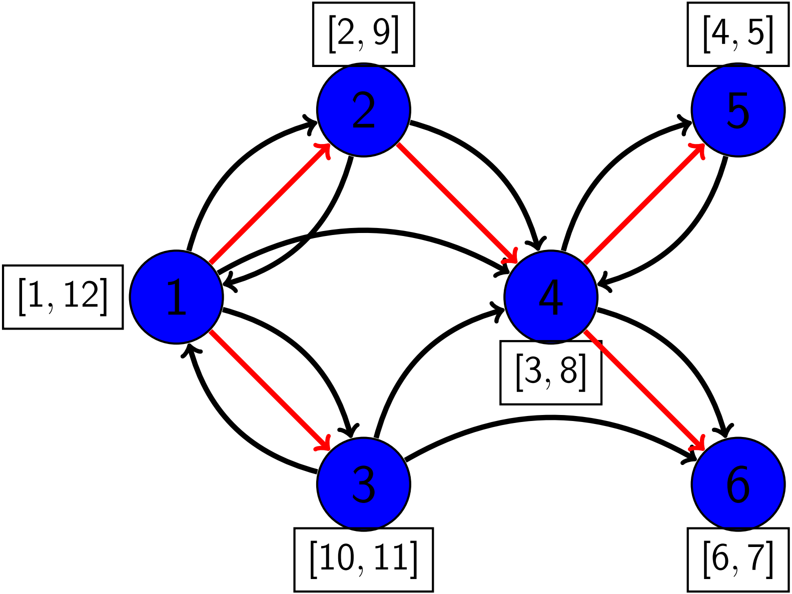
Busca em Profundidade - Exemplo



Busca em Profundidade - Exemplo



Busca em Profundidade - Exemplo



Busca em Profundidade - Produtos

- Grafo de predecessores $G_\pi = (V, E_\pi)$.
- Os vetores de tempo de descoberta e de tempo de visita.
- Complexidade:
  Lista de Adjacências: $\Theta(|V| + |E|)$
  Matriz de Adjacências: $\Theta(|V|^2)$

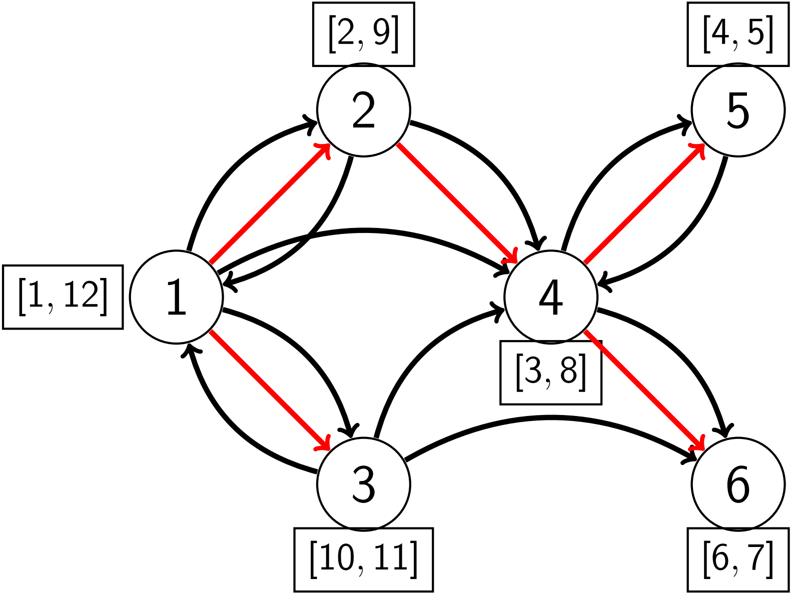
Busca em Profundidade - Estrutura de Parêntesis

Teorema dos Parêntesis

Em qualquer busca em profundidade de um grafo $G = (V, E)$, para quaisquer dois vértices v, u, uma e apenas uma das condições abaixo ocorre:

i. $[i[v], f[v]] \cap [i[u], f[u]] = \emptyset$ e u e v não descentes um do outro em G_π;
ii. $[i[v], f[v]] \cap [i[u], f[u]] = [i[v], f[v]]$ e v é descendente de u em G_π;
iii. $[i[v], f[v]] \cap [i[u], f[u]] = [i[u], f[u]]$ e u é descendente de v em G_π;

Estrutura de Parêntesis - Exemplo



Busca em Profundidade - Teorema do Caminho Branco

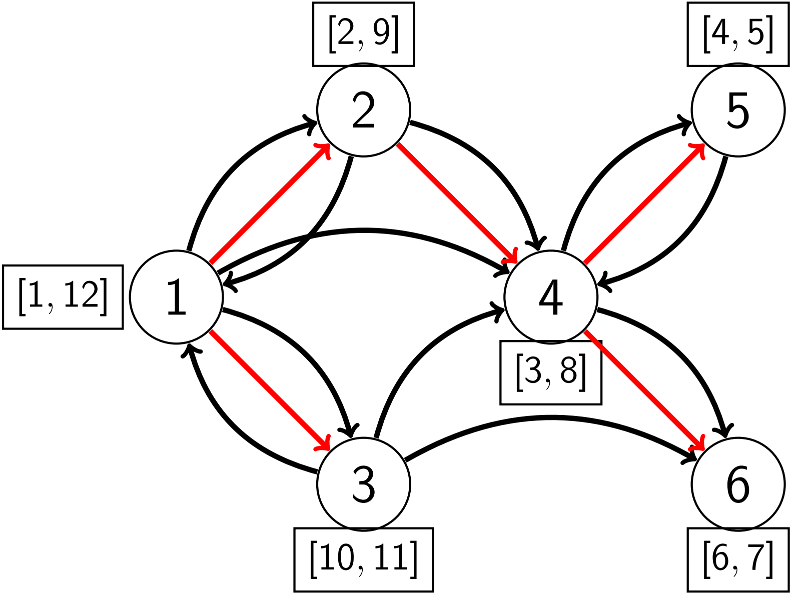
Teorema do Caminho Branco

Em G_π, um vértice v é descendente de um vértice u se e somente se no tempo $i[u]$ existe um caminho de vértices brancos de u até v.

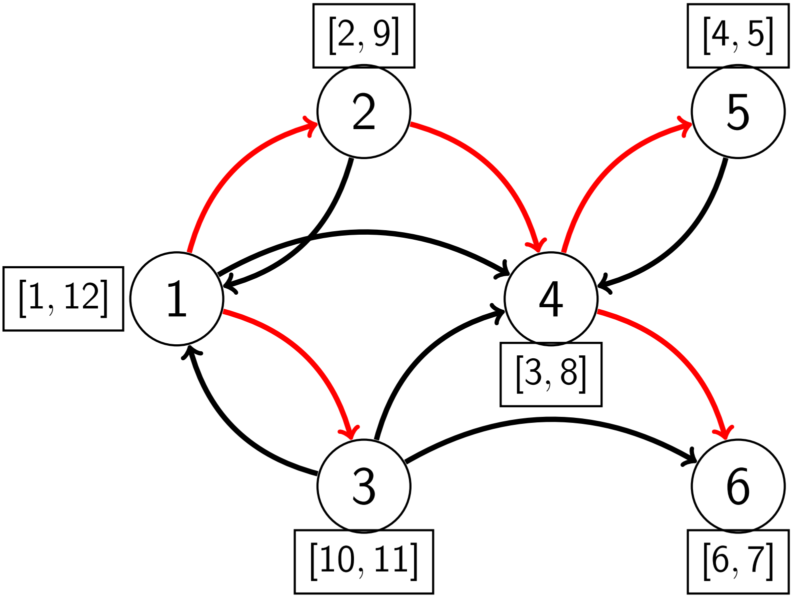
Busca em Profundidade - Classificação das Arestas

- Podemos utilizar o grafo G_π através de uma busca em profundidade para classificar as arestas de G.
- Aresta uv de Árvore: $vu \in G_\pi$.
- Aresta uv de Volta: liga u a um ancestral.
- Aresta uv de Avanço: liga u a um descendente.
- Aresta uv de Passagem: as demais.

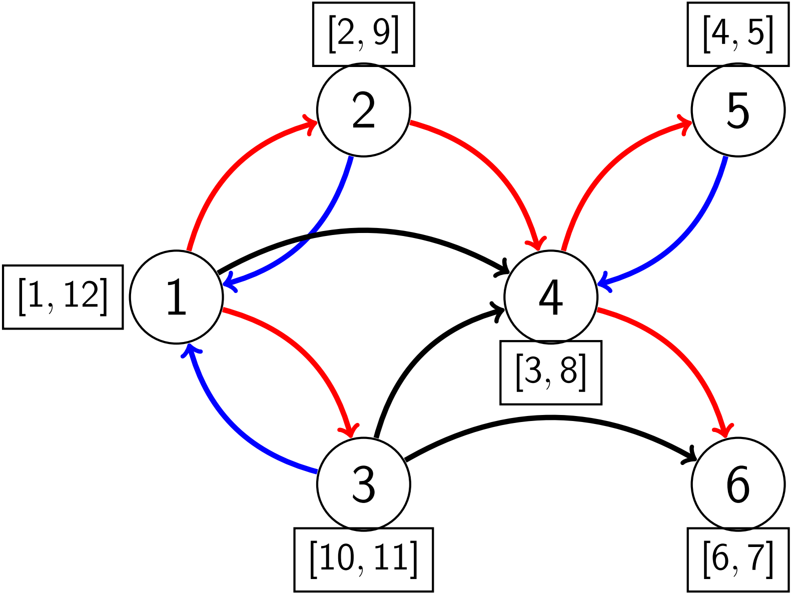
Classificação das Arestas



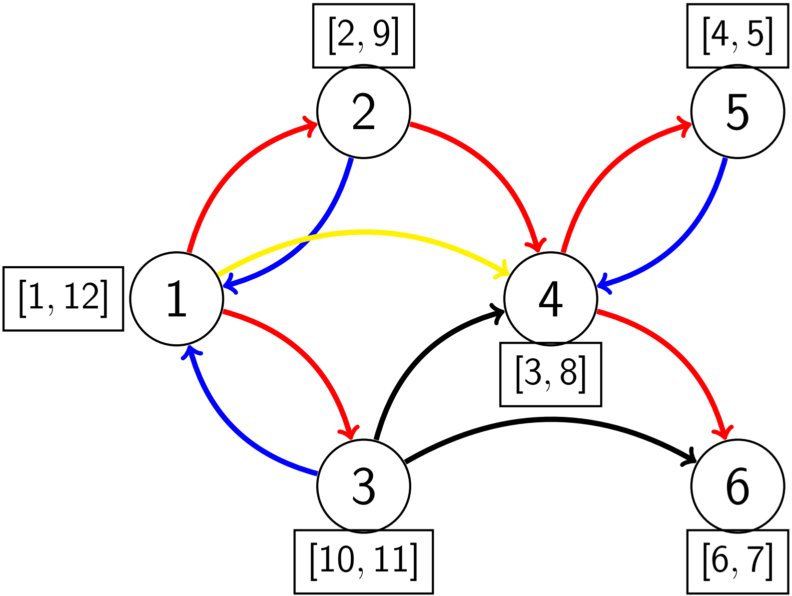
Classificação das Arestas



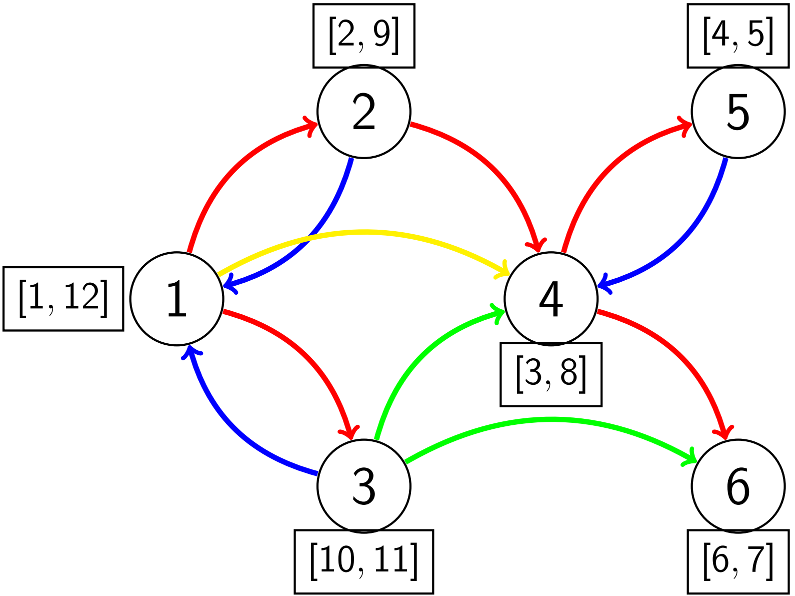
Classificação das Arestas



Classificação das Arestas



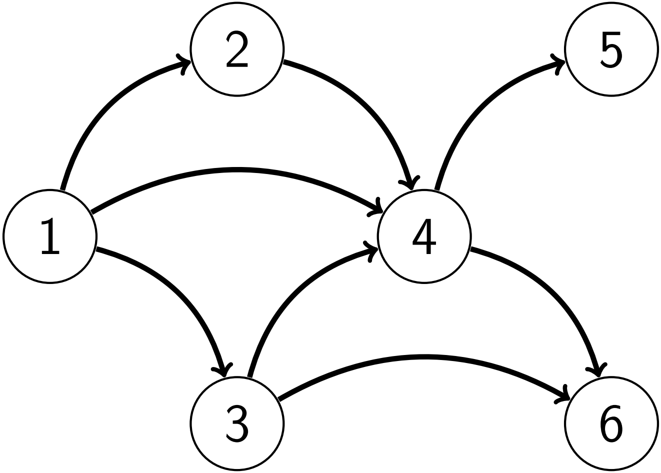
Classificação das Arestas



Ordenação Topológica

- Dado um grafo orientado $G = (V, A)$.
- Assuma que este grafo não possui circuitos orientados.
- Chamamos esse tipo de grafo orientado DAG.
- Esse tipo de grafo representa uma ordem parcial.
- Queremos uma ordem total que respeite essa ordem parcial.

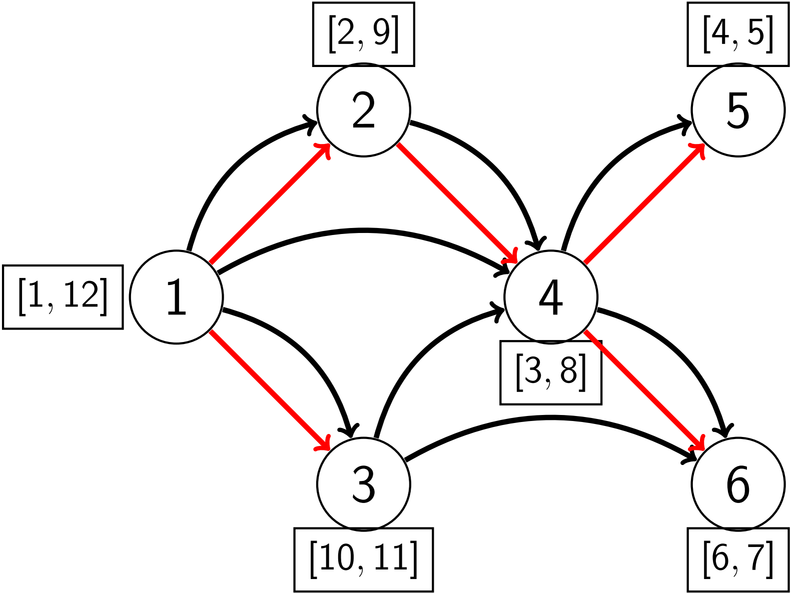
Ordenação Topológica - Exemplo



Ordenação Topológica

Entrada: Grafo $G = (V, s)$.
Use DFS para calcular os tempos $f[v]$;
Ordene inversamente $V(G)$ por $f[v]$;
´
Algoritmo 3: Ordenacao_Topologica(G)

Ordenação Topológica - Exemplo



Componentes Fortemente Conexas

- Dado um grafo orientado $G = (V, A)$.
- dizemos que um par de vértices u, v é fortemente conectado se existe um caminho de u para v e vice e versa.
- Grafo Fortemente Conexo: se todo par de vértice é fortemente conectado.
- Componente Fortemente Conexa: conjunto maximal de vértices que induz um grafo fortemente conexo.

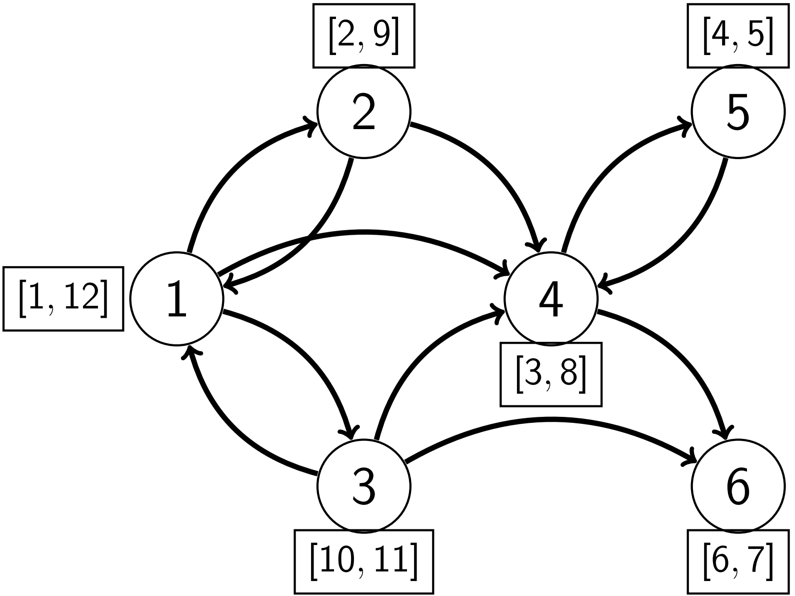
Componentes Fortemente Conexas

- Um grafo orientado pode ser decomposto em componentes fortemente conexas.
- Defina G^T como
  - $V(G^T) = V(G)$
  - $A(G^T) =$ conjunto de arcos de G com a direção trocada.

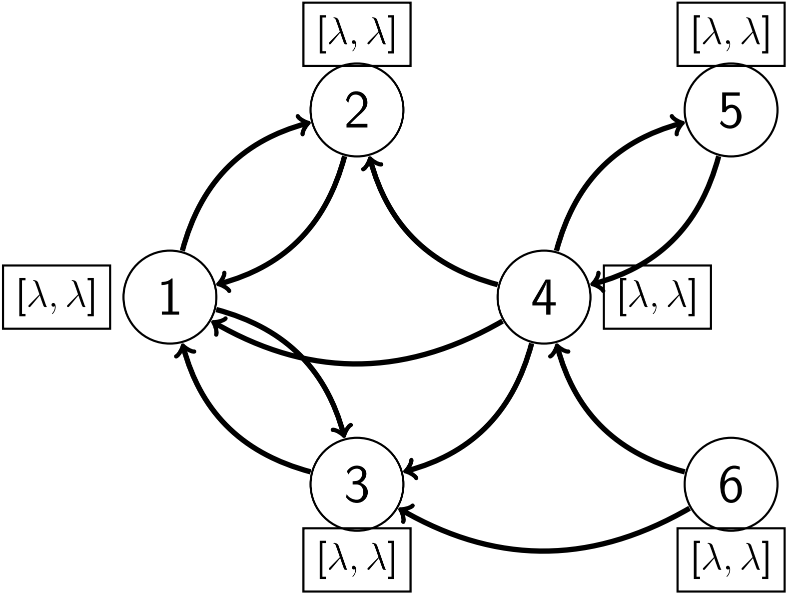
Componentes Fortemente Conexas

Entrada: Grafo $G = (V, s)$.
Use DFS para calcular os tempos $f[v]$;
Compute G^T;
Use DFS em G^T, mas com os vértices ordenados em forma decrescente por valor de $f[v]$;
´
Algoritmo 4: Componentes_Fortemente_Conexas(G)

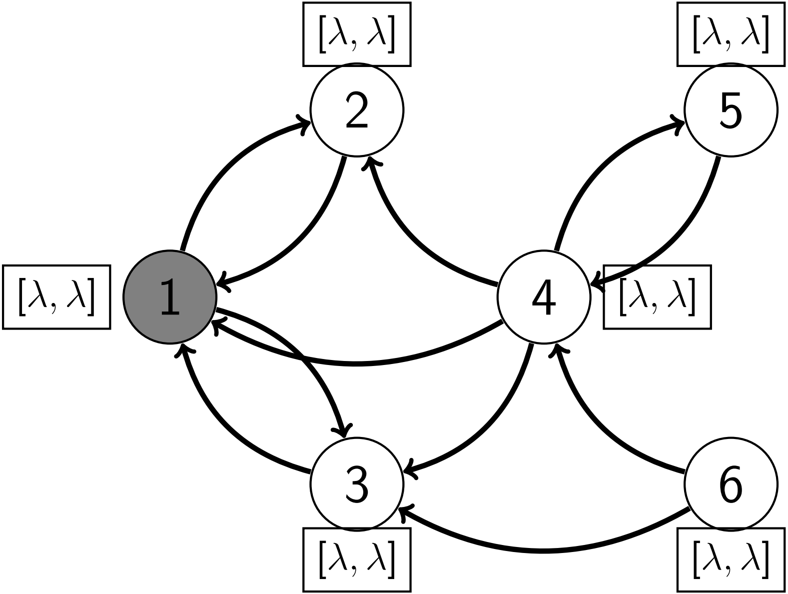
Componentes Fortemente Conexas



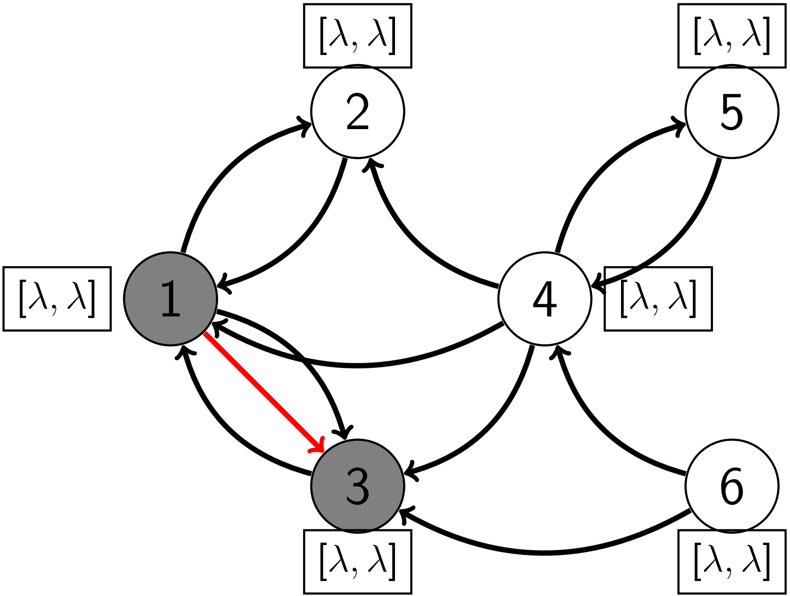
Componentes Fortemente Conexas



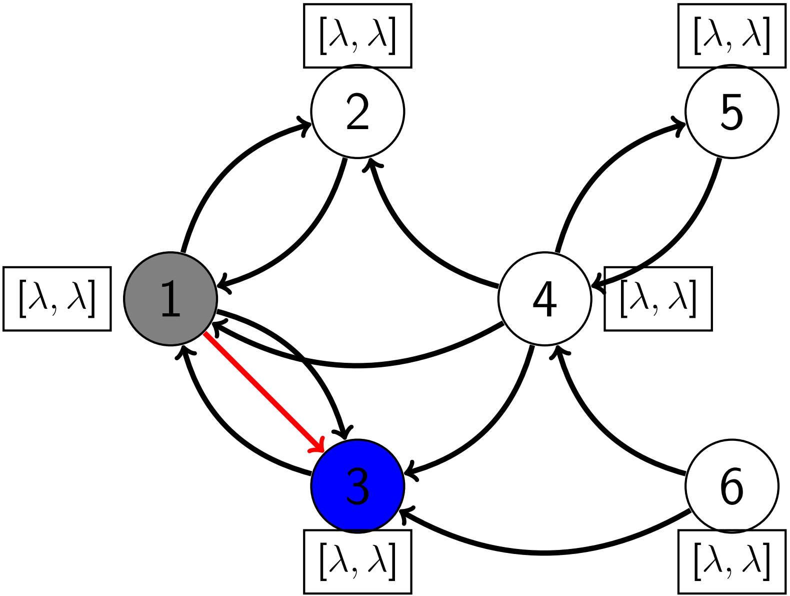
Componentes Fortemente Conexas



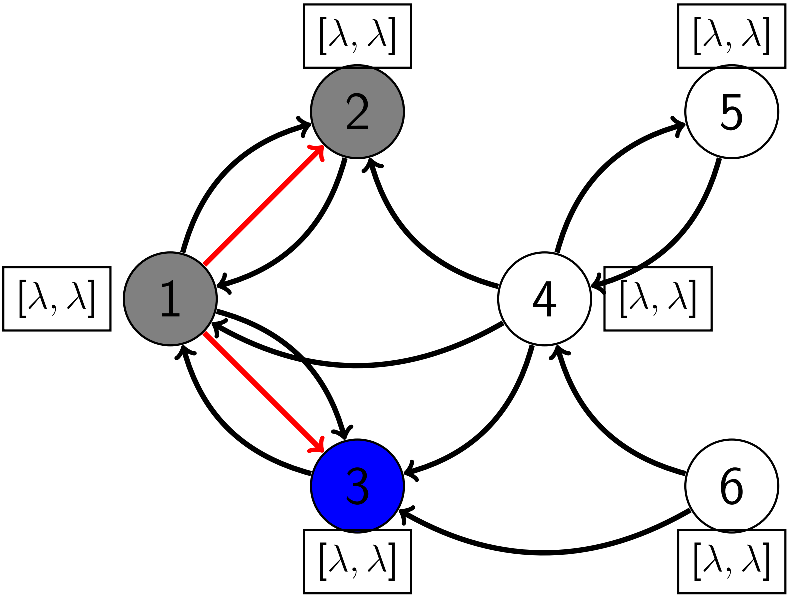
Componentes Fortemente Conexas



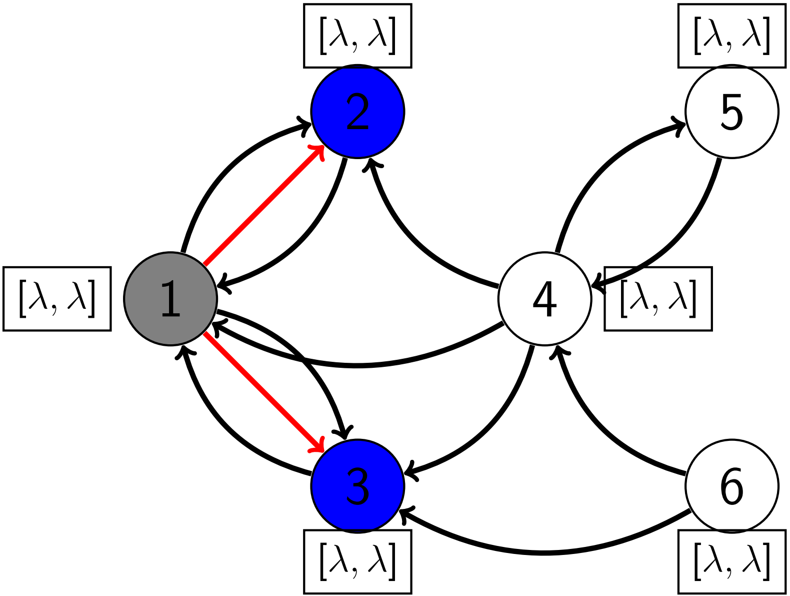
Componentes Fortemente Conexas



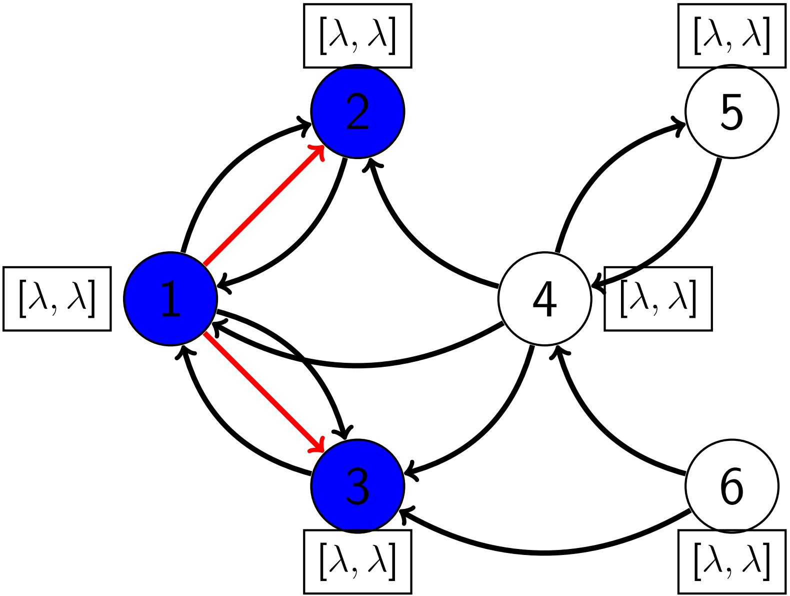
Componentes Fortemente Conexas



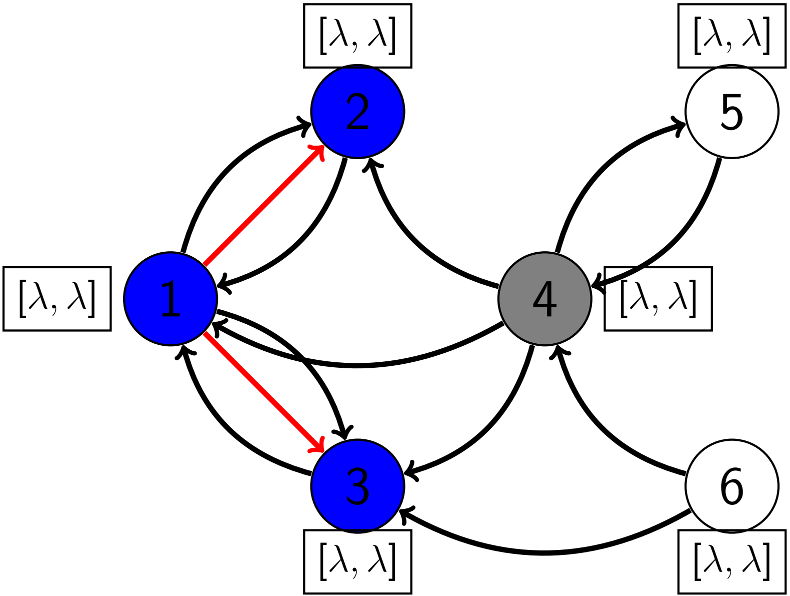
Componentes Fortemente Conexas



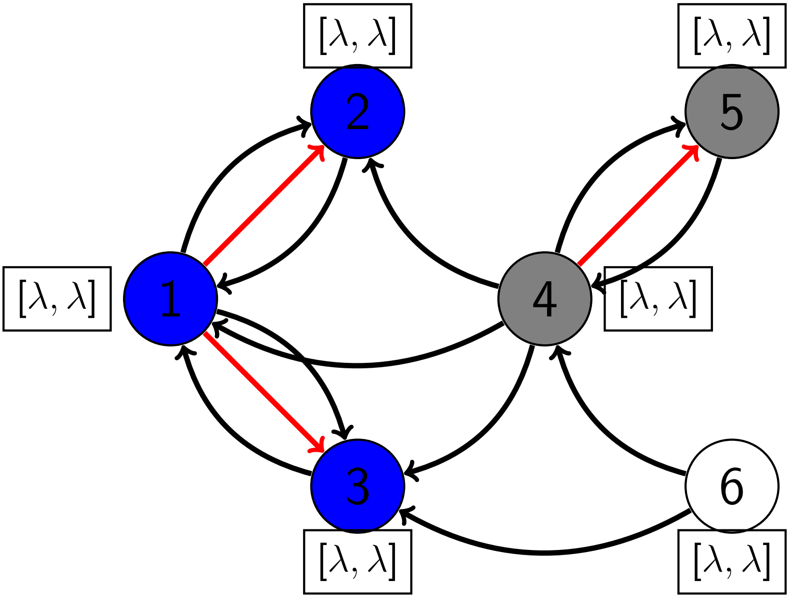
Componentes Fortemente Conexas



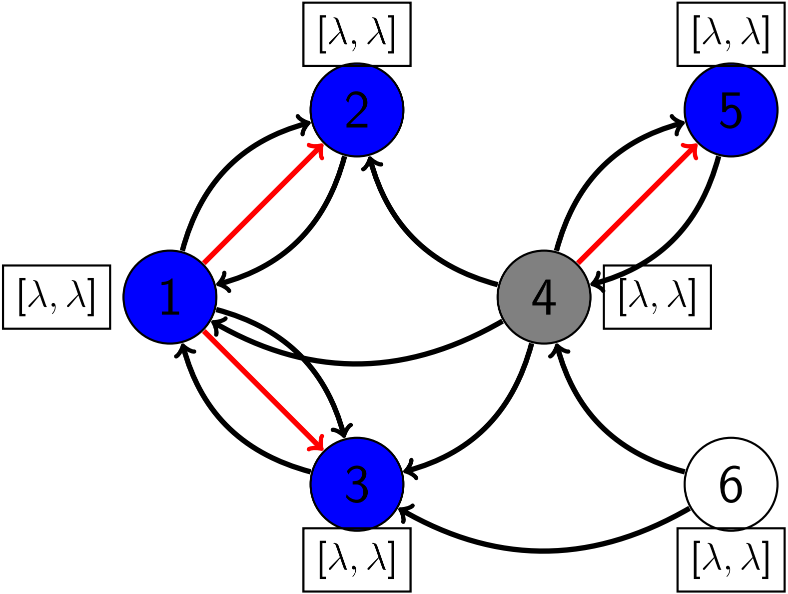
Componentes Fortemente Conexas



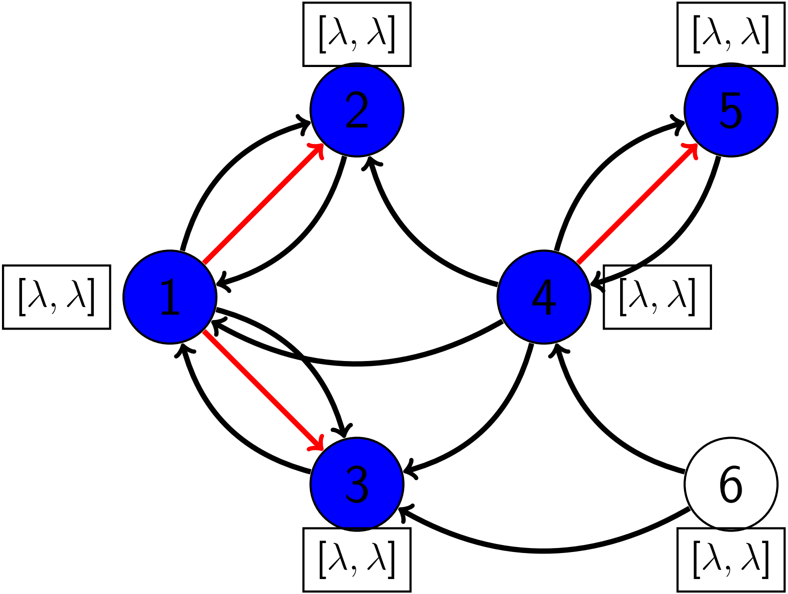
Componentes Fortemente Conexas



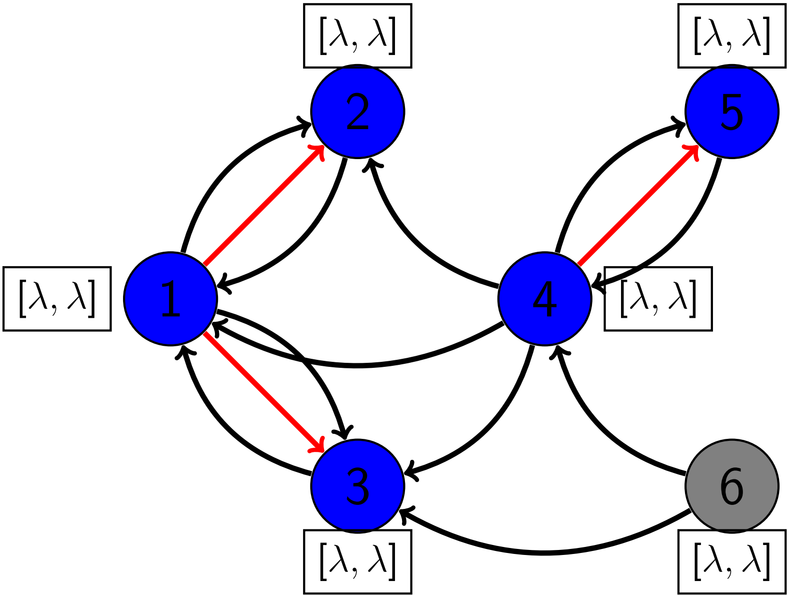
Componentes Fortemente Conexas



Componentes Fortemente Conexas



Componentes Fortemente Conexas



Componentes Fortemente Conexas