

# ESTRUTURA DE DADOS

## Grafos

### Busca em Largura

Professor Mestre Igor de Moraes Sampaio  
[igor.sampaio@ifsp.edu.br](mailto:igor.sampaio@ifsp.edu.br)



# Busca em Largura



# Busca em Largura

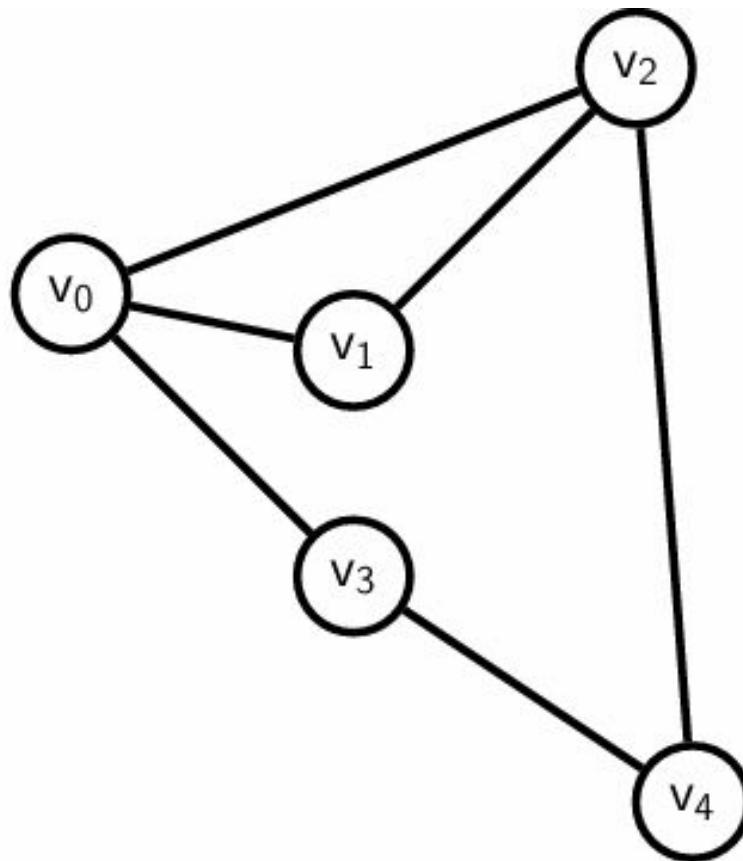
---

- A Busca em Largura (ou do inglês Breadth-First Search BFS) é um algoritmo ou estratégia que pode ser utilizada para realizar uma busca ou travessia estruturas como árvores e grafos.
- Em grafos, o algoritmo começa a partir de um vértice e explora, inicialmente, todos os seus vizinhos, depois os vizinhos dos vizinhos e assim sucessivamente.

# Ideia

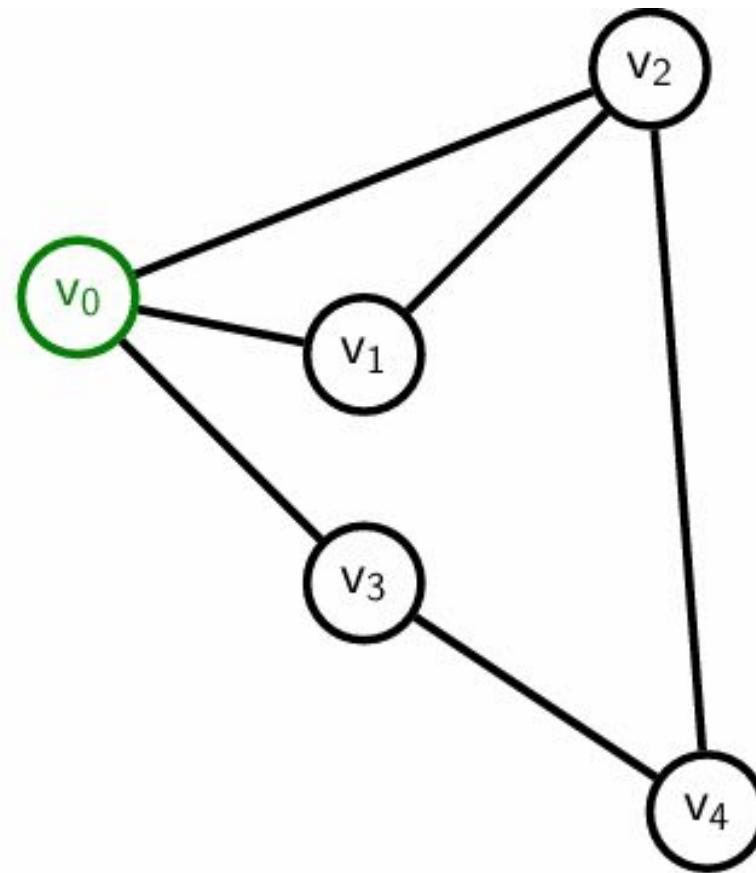
# Busca

---



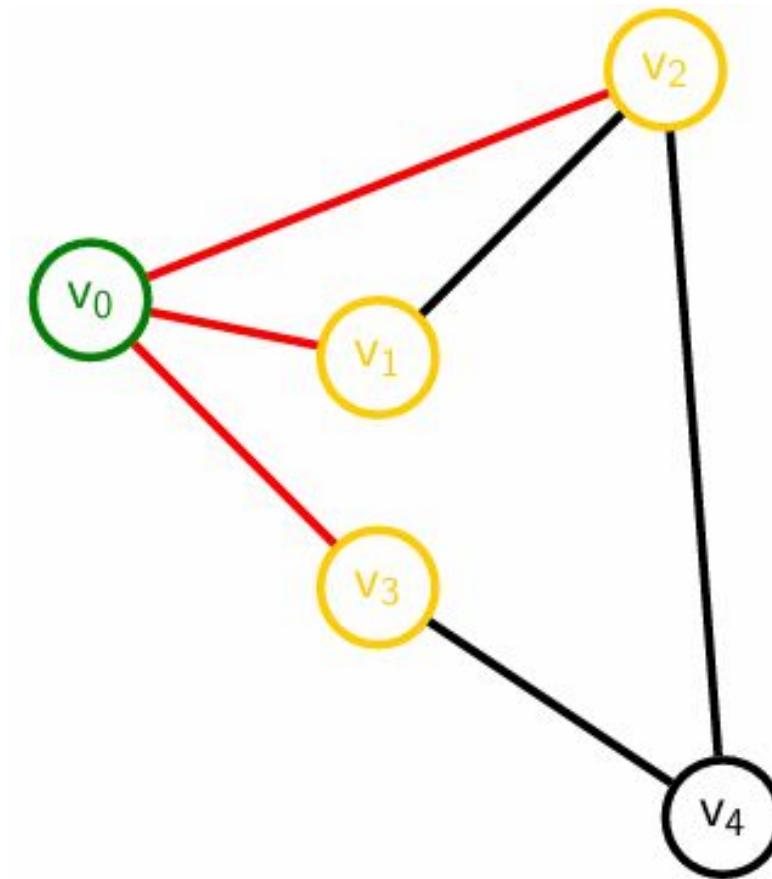
# Busca

---



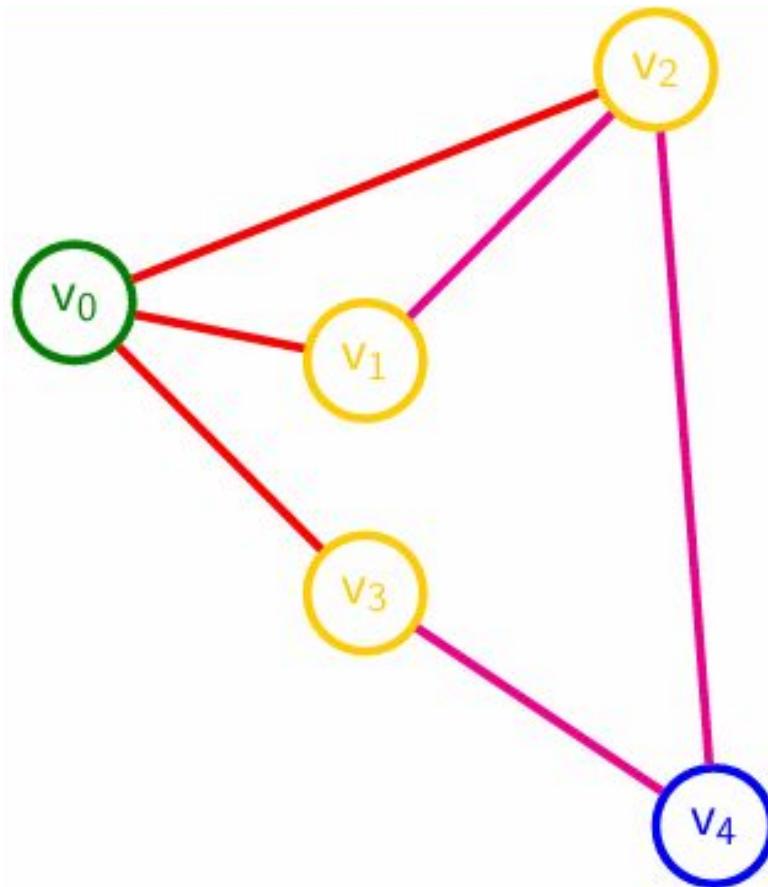
# Busca

---



# Busca

---





# Busca em Largura

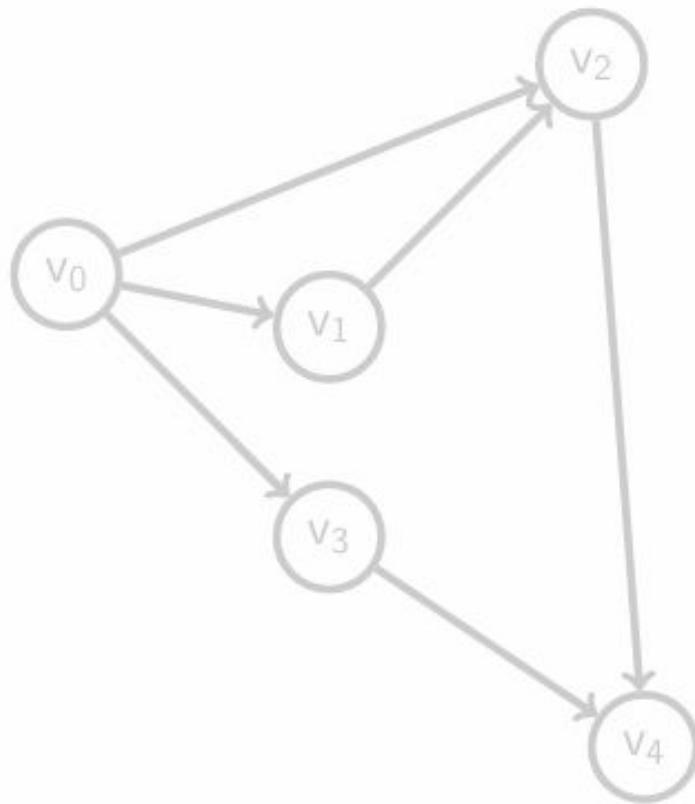
---

- Em termos de implementação, temos que visitar um dos vizinhos de cada vez.
- Mas, diferentemente da Busca em Profundidade, primeiro visitaremos todos os vizinho do nó atual para só depois visitarmos os vizinhos dos vizinhos.
- A cada vértice visitado, colocaremos numa fila seus vizinhos, para que sejam visitados apenas após todos aqueles que já estão na fila.

# Processo

# Busca

---

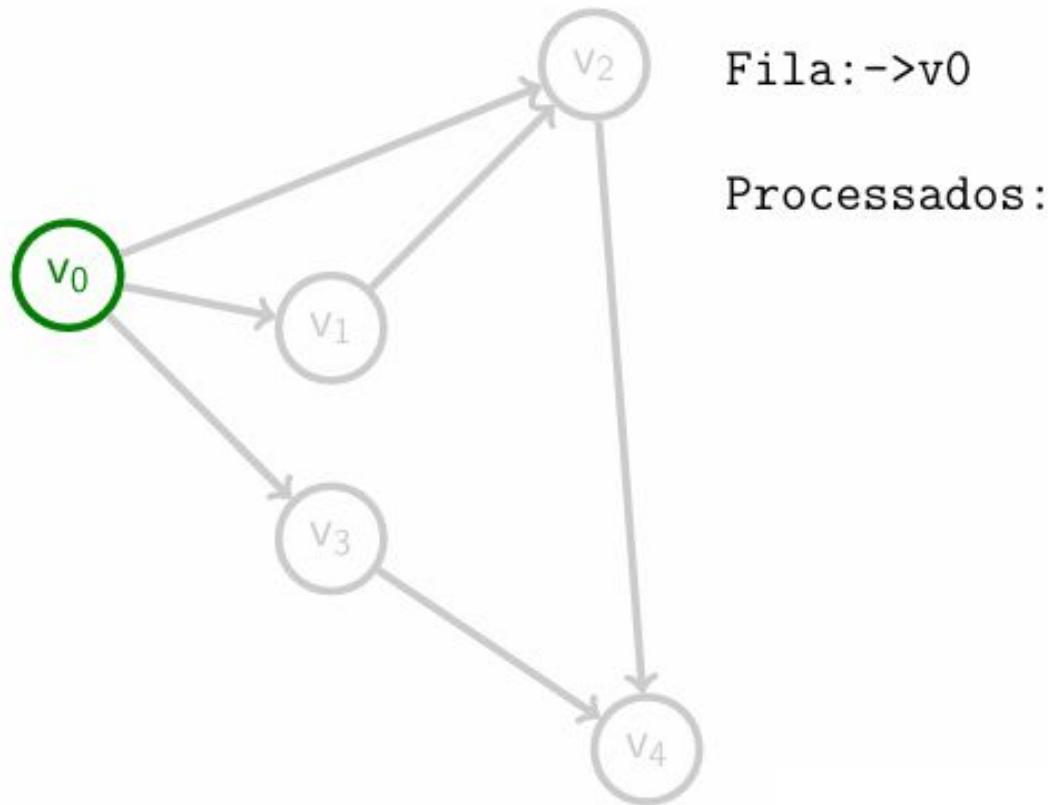


Fila:

Processados:

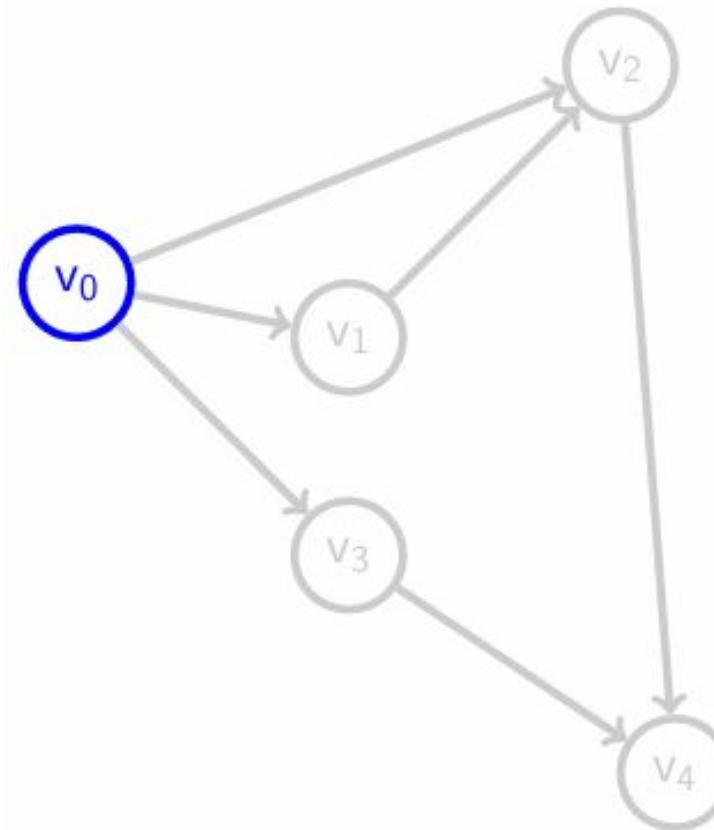
# Busca

---



# Busca

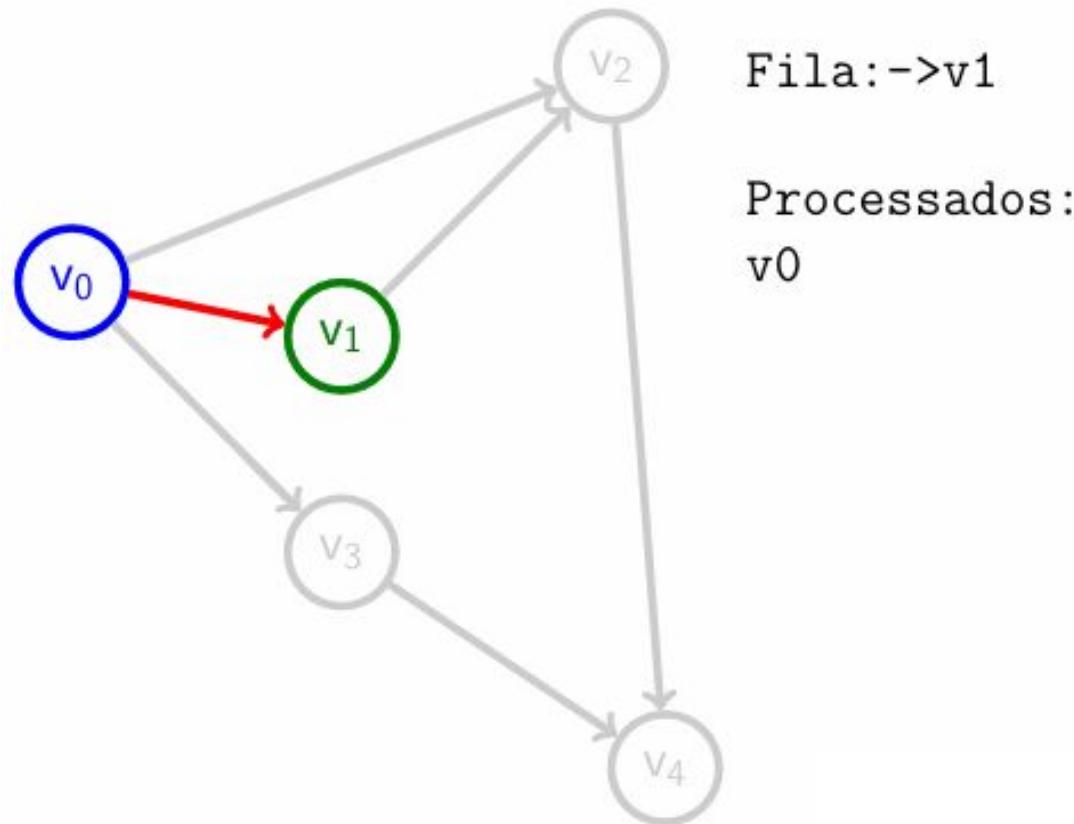
---



Fila:

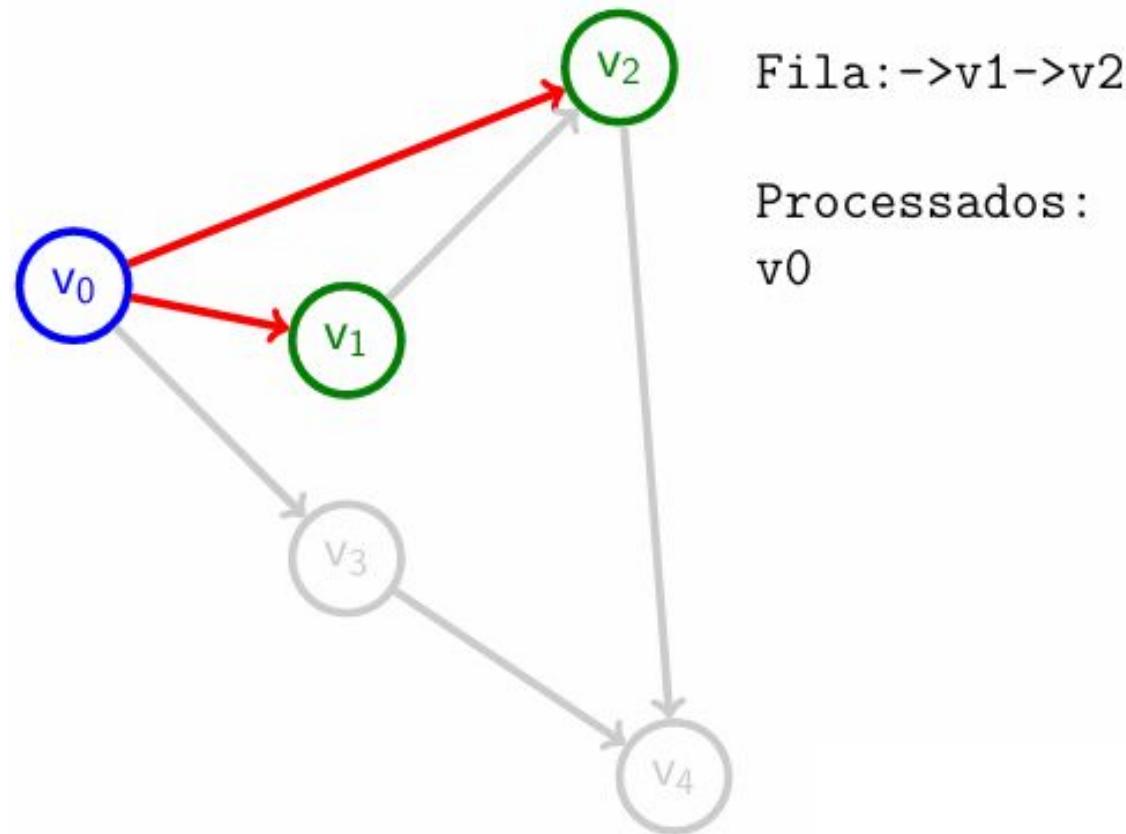
Processados:  
v0

# Busca

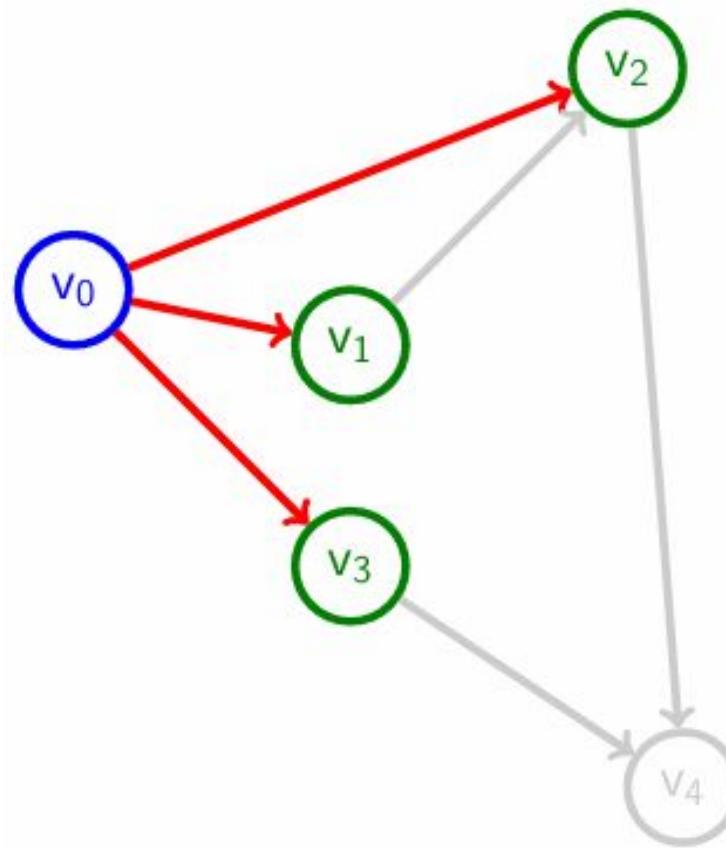


# Busca

---



# Busca

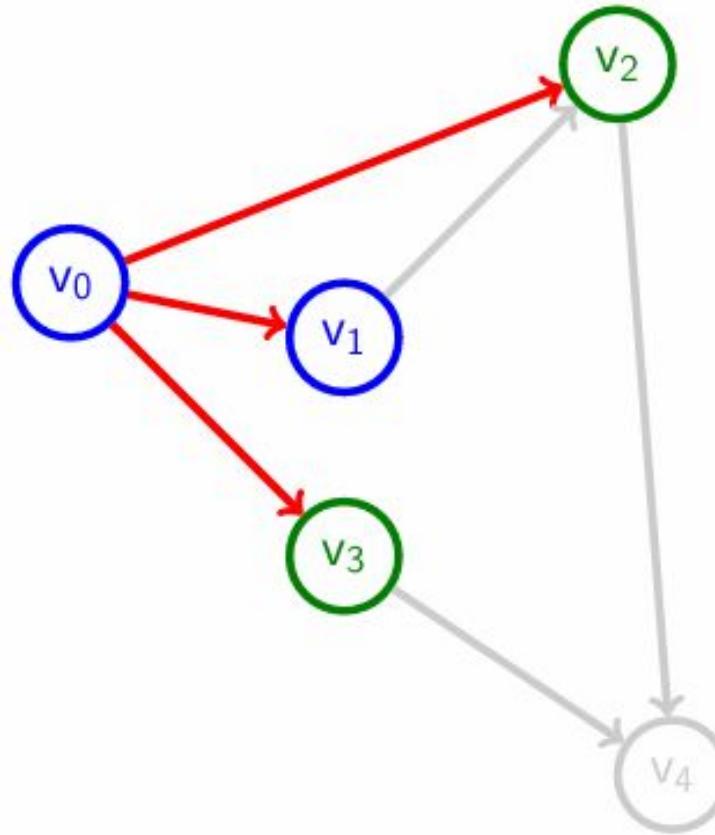


Fila: $\rightarrow v_1 \rightarrow v_2 \rightarrow v_3$

Processados:  
v0

# Busca

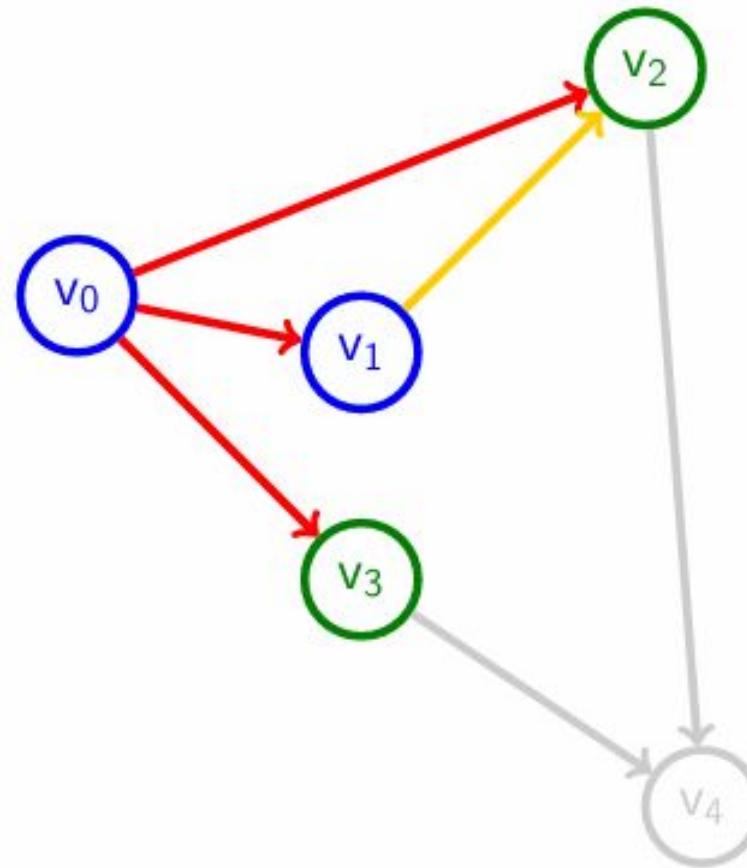
---



Fila: ->v2->v3

Processados:  
v0 v1

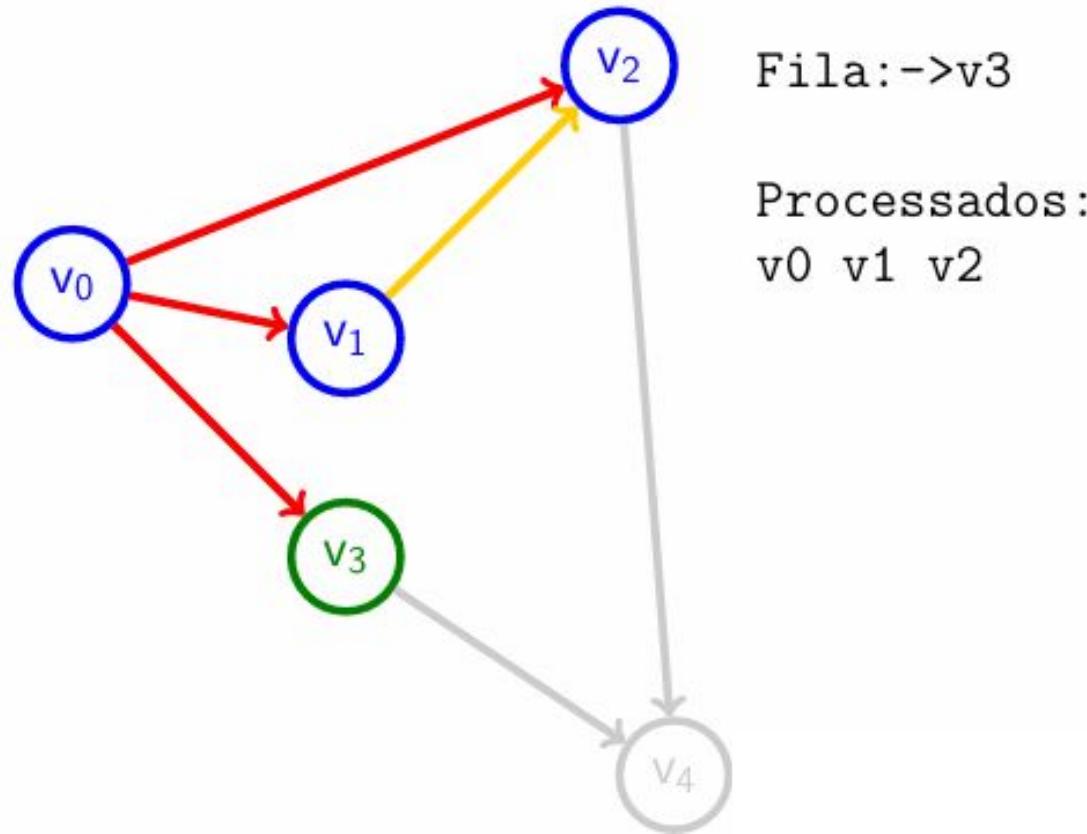
# Busca



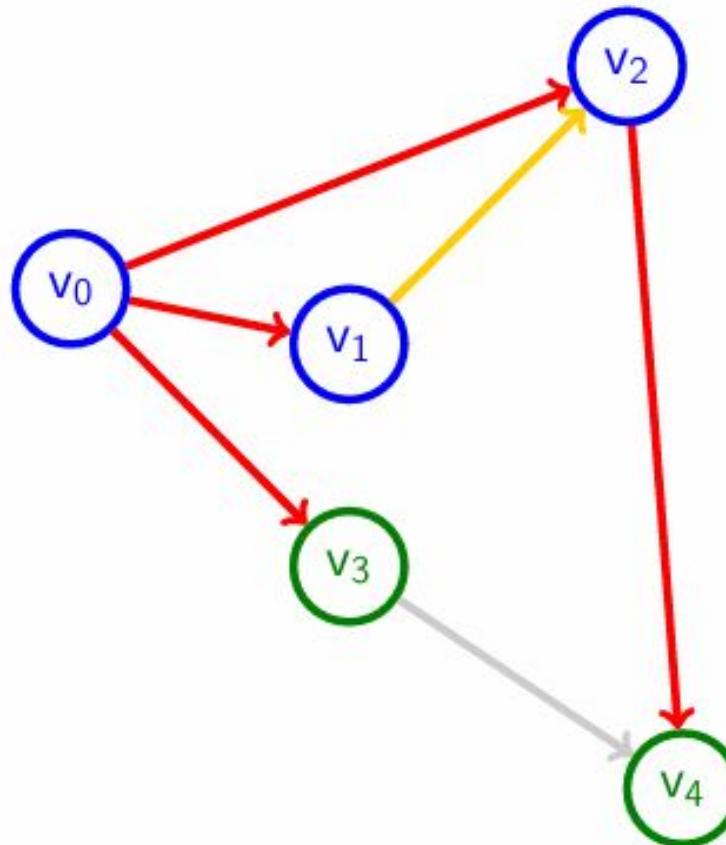
Fila: ->v2->v3

Processados:  
v0 v1

# Busca



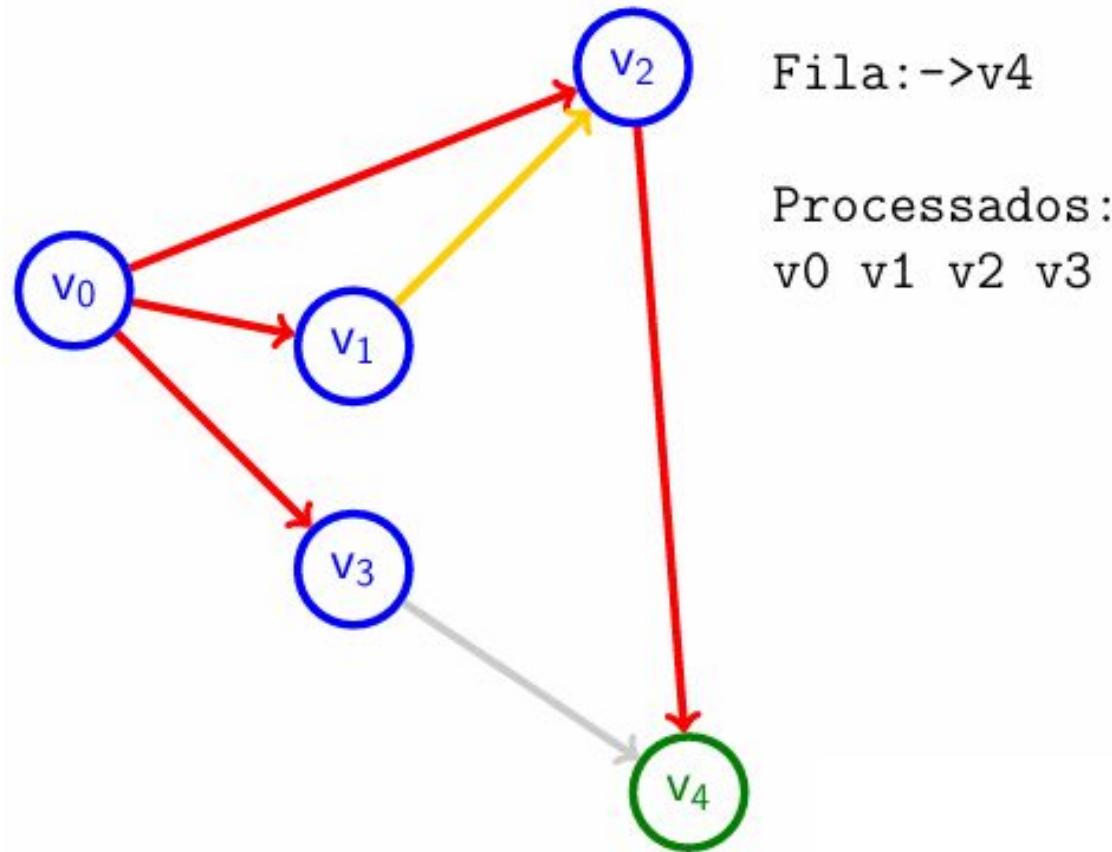
# Busca



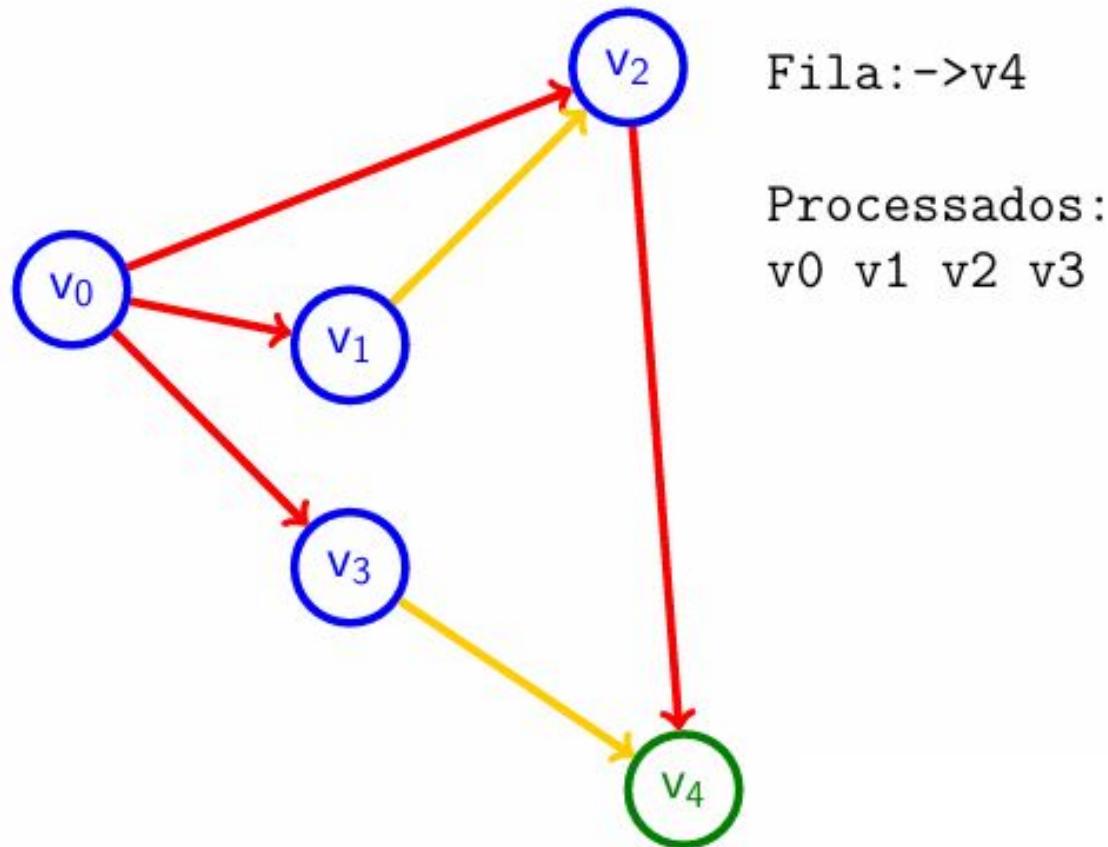
Fila: ->v3->v4

Processados:  
v0 v1 v2

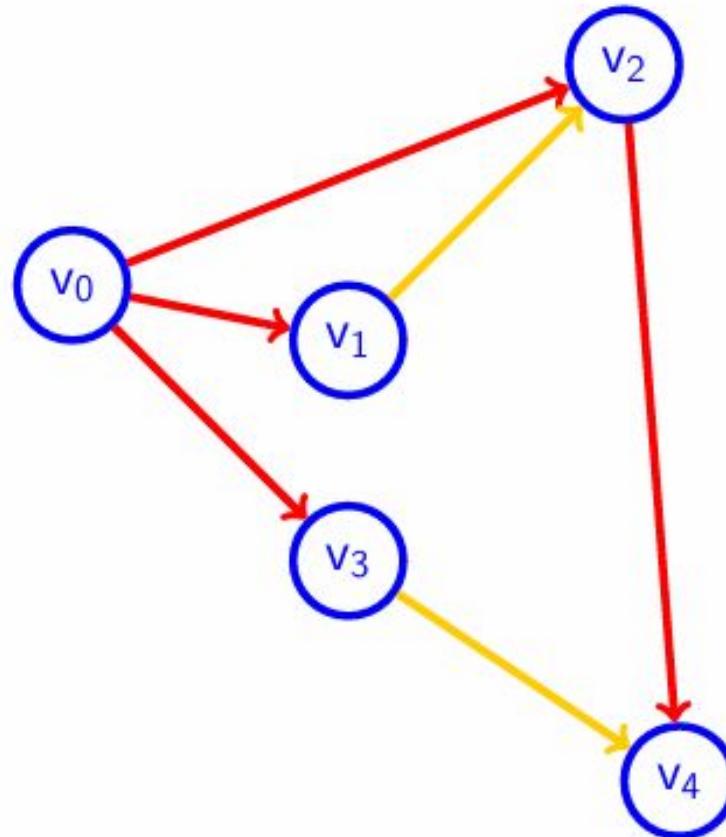
# Busca



# Busca



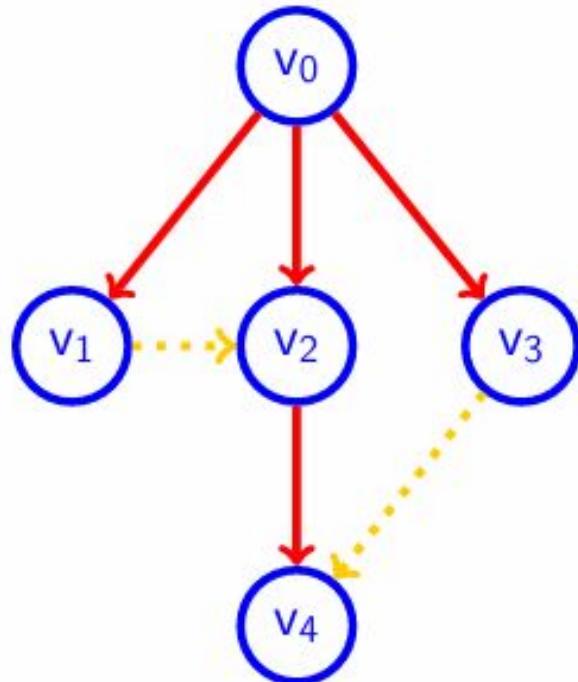
# Busca



Fila:

Processados:

v0 v1 v2 v3 v4



- A árvore produzida com as arestas a partir das quais visitamos cada vértice pela primeira vez permite a descoberta do caminho de menor comprimento entre a raiz (vértice inicial) e os demais.



# Conceito

---

- Precisaremos 'enfileirar' os vizinhos de cada vértice visitado.
- Usar a pilha de recursão não funcionaria aqui! Precisamos de uma fila.
- Fila é aquela estrutura de dados em que as inserções ocorrem no final e as exclusões no início.

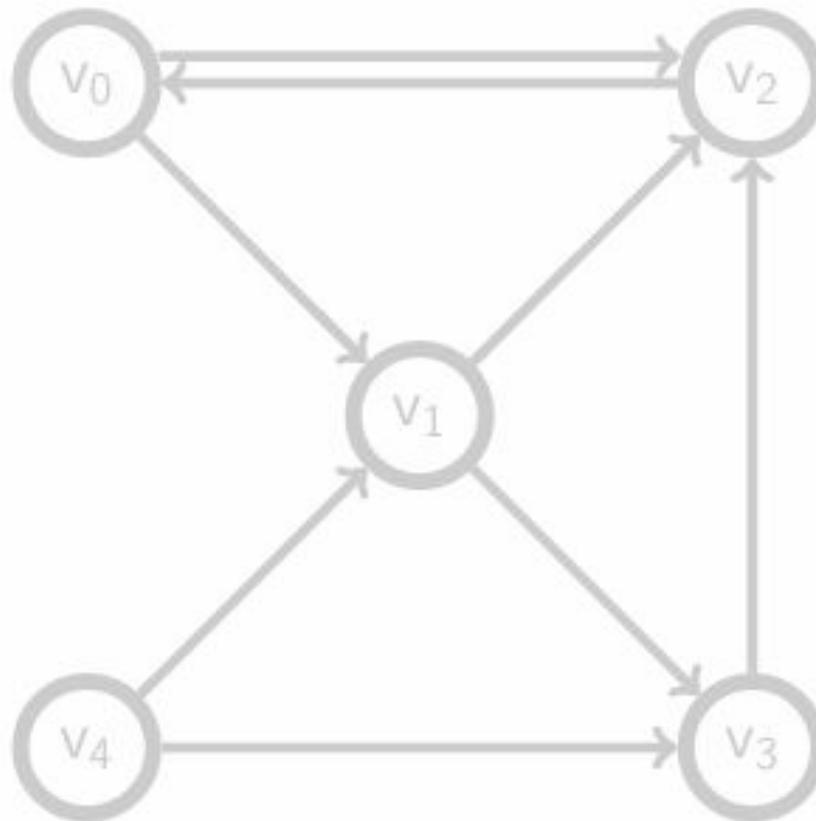
# Aplicações

## Para que serve a Busca em Largura?

- Pode ser usada (ou adaptada) para resolver diferentes problemas:
  - Encontrar o caminho de menor comprimento entre dois vértices (em termos de número de arestas percorridas) - solução de labirinto
  - Encontrar componentes conexos
  - Verificar se um grafo é bipartido
  - Auxiliar no problema de fluxo máximo
  - ...

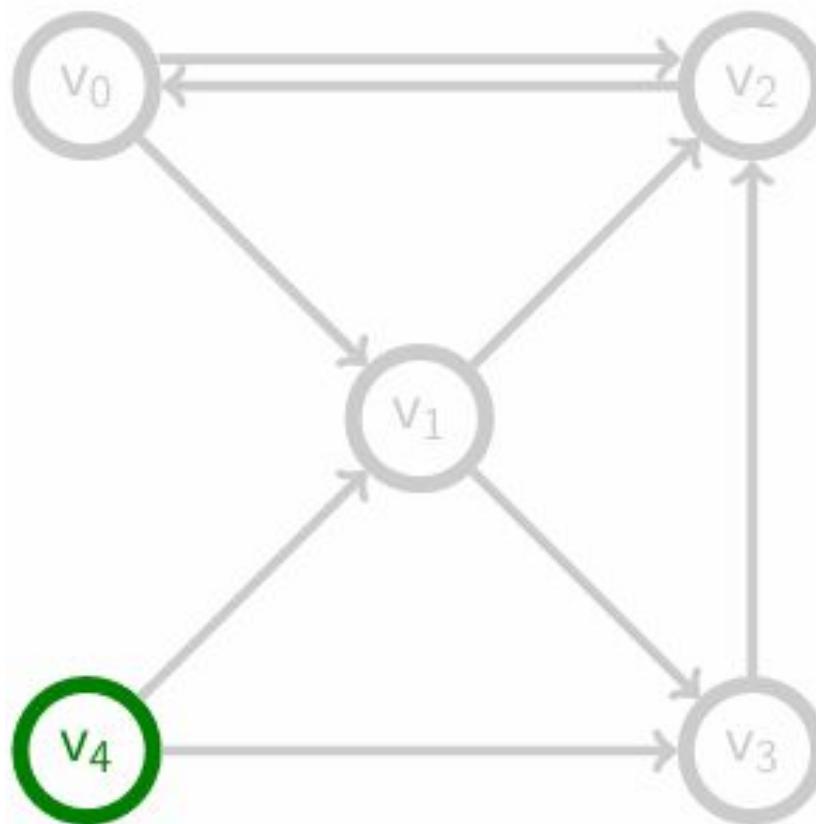
# Labirinto

---



Início  $v_4$ , destino  $v_0$

# Labirinto

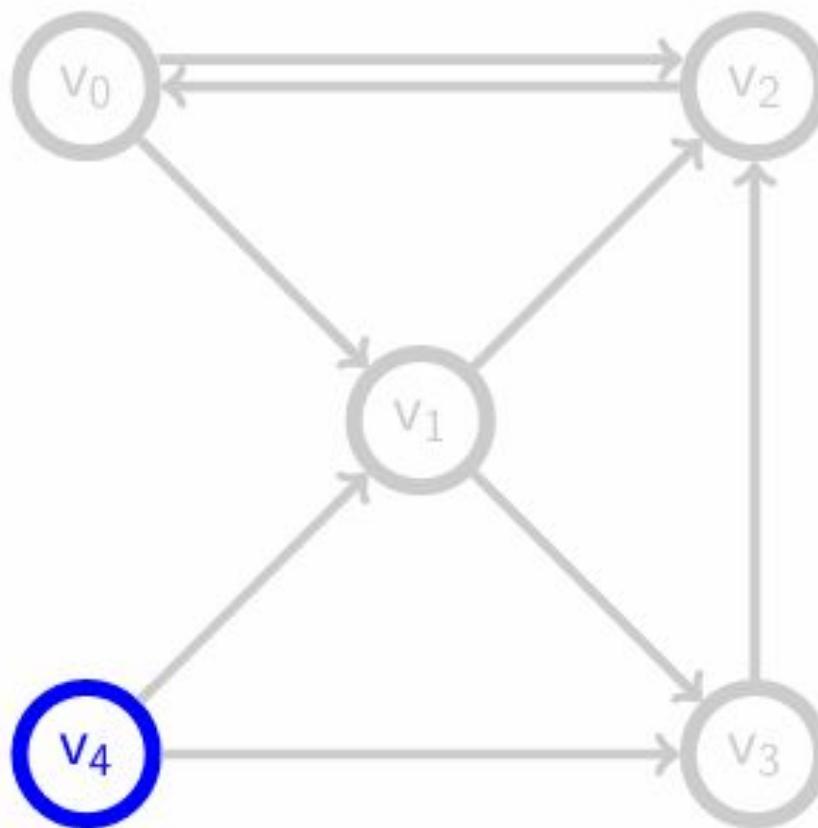


Início  $v_4$ , destino  $v_0$

Fila: -> $v_4$

Processados:

# Labirinto

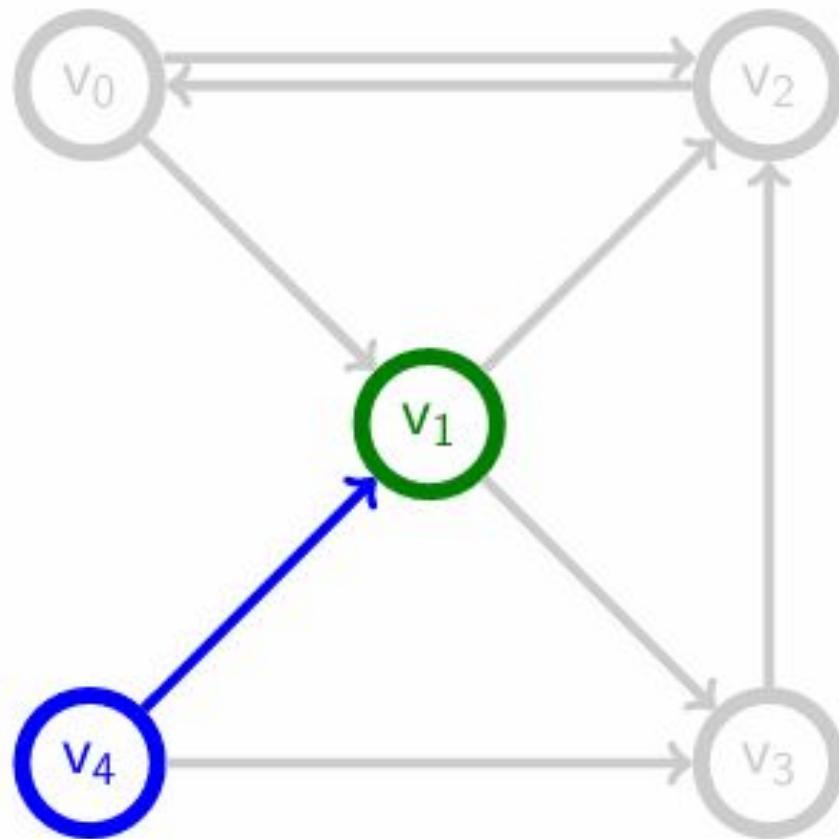


Início  $v_4$ , destino  $v_0$

Fila:

Processados:  
 $v_4$

# Labirinto

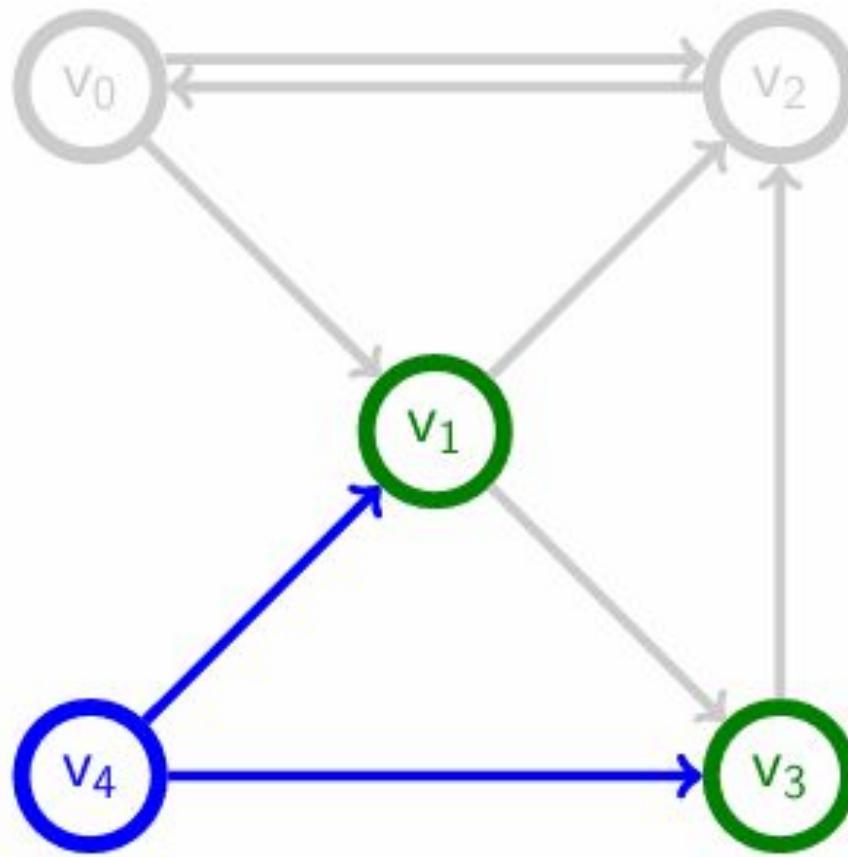


Início  $v_4$ , destino  $v_0$

Fila: -> $v_1$

Processados:  
 $v_4$

# Labirinto

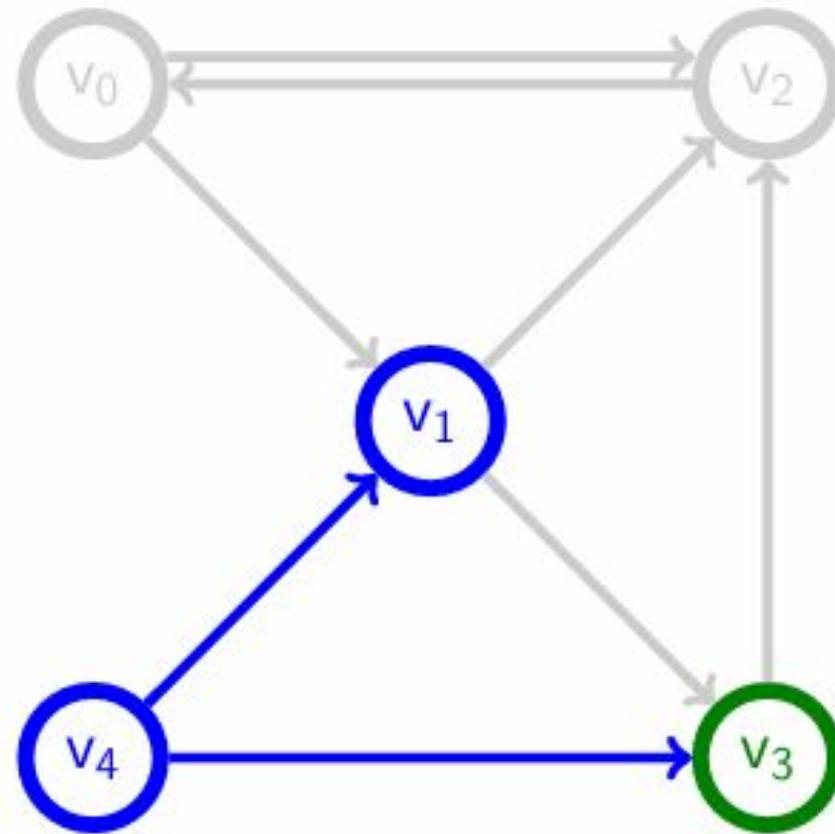


Início  $v_4$ , destino  $v_0$

Fila: -> $v_1$ -> $v_3$

Processados:  
 $v_4$

# Labirinto

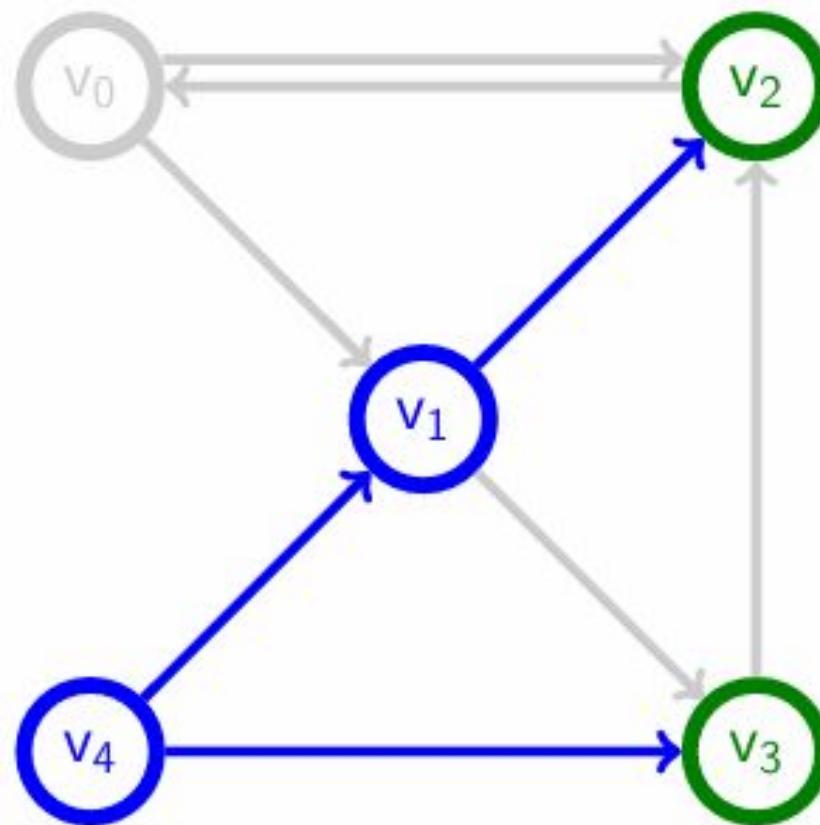


Início  $v_4$ , destino  $v_0$

Fila: -> $v_3$

Processados:  
 $v_4 \ v_1$

# Labirinto

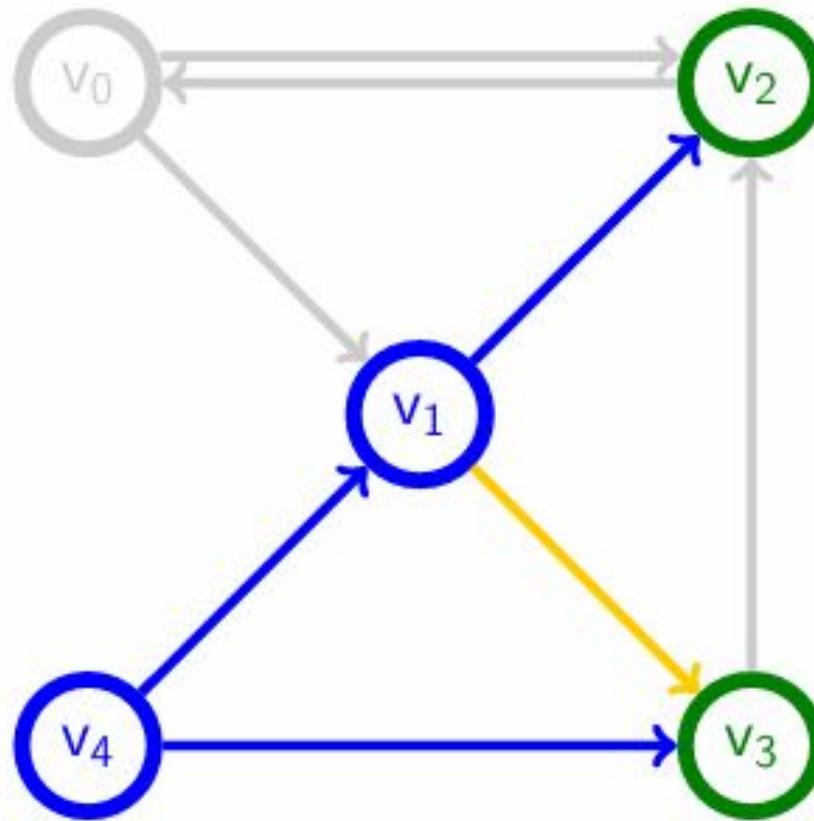


Início  $v_4$ , destino  $v_0$

Fila: -> $v_3$ -> $v_2$

Processados:  
 $v_4$   $v_1$

# Labirinto

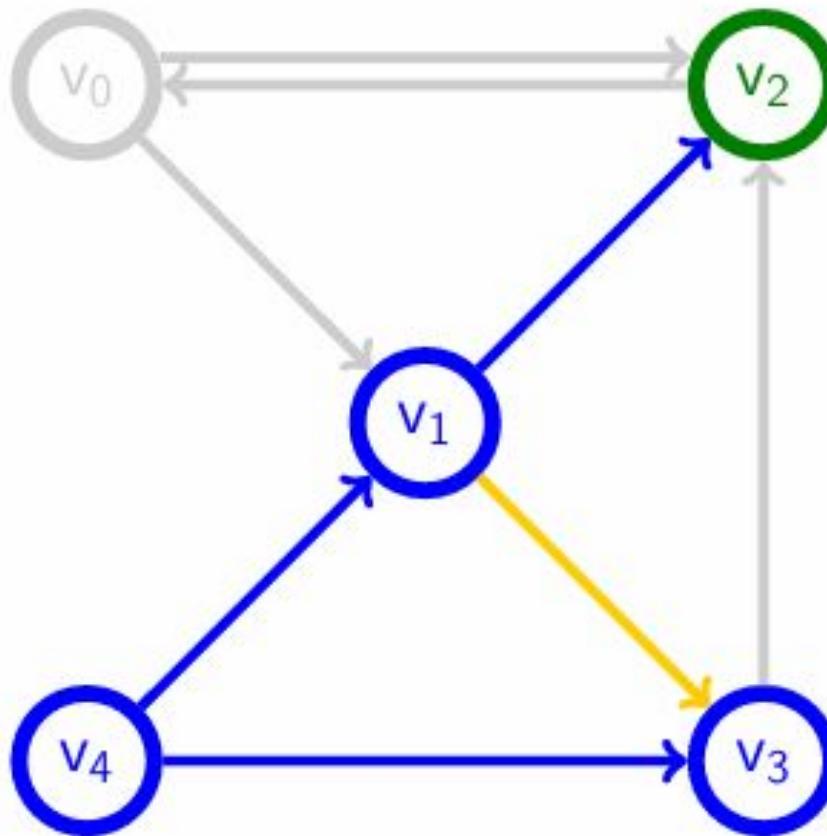


Início  $v_4$ , destino  $v_0$

Fila: -> $v_3$ -> $v_2$

Processados:  
 $v_4$   $v_1$

# Labirinto

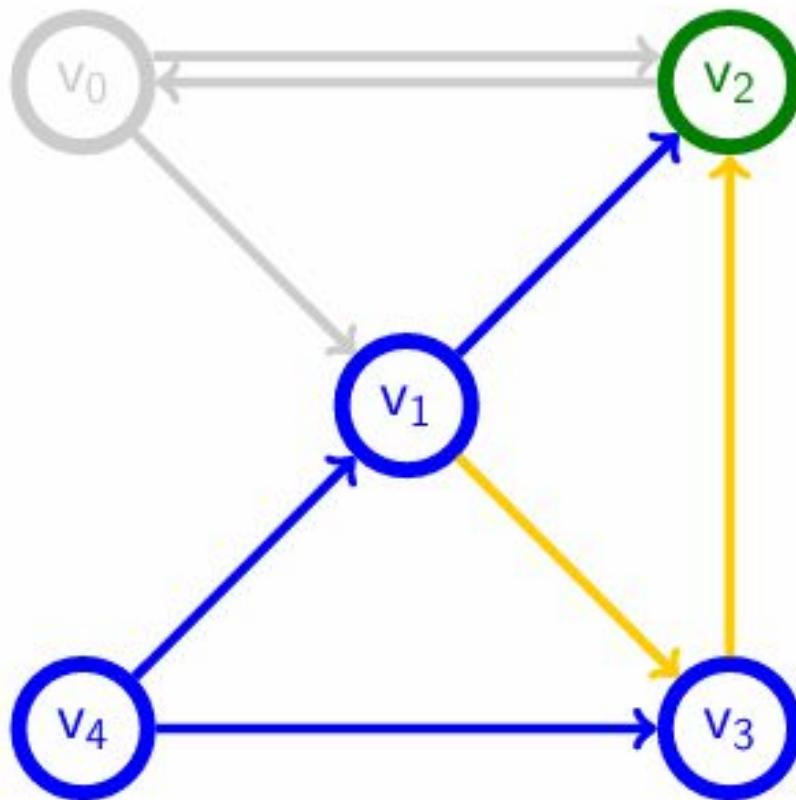


Início  $v_4$ , destino  $v_0$

Fila: -> $v_2$

Processados:  
 $v_4 \ v_1 \ v_3$

# Labirinto

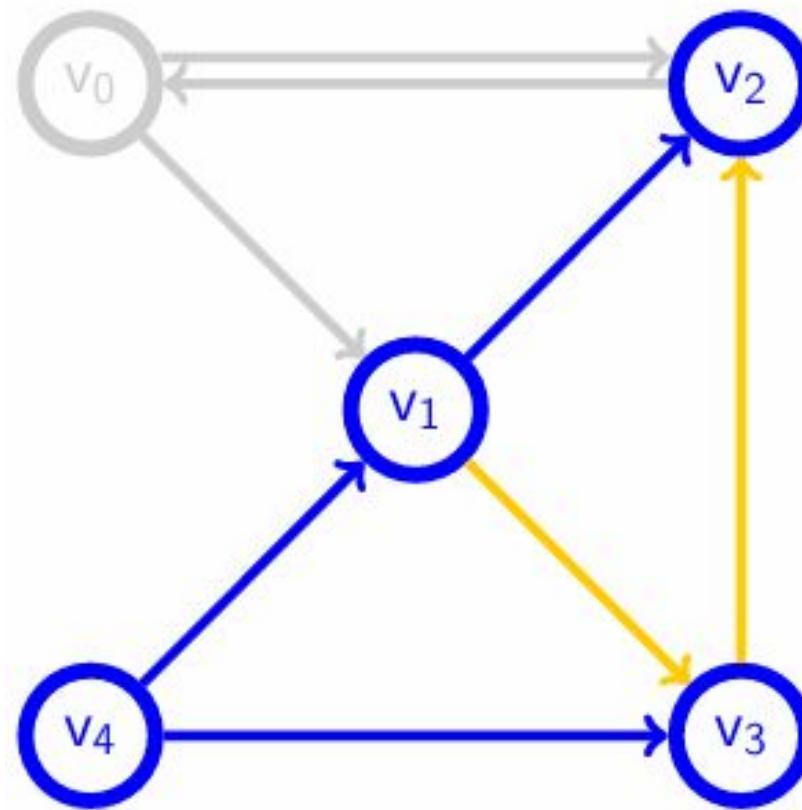


Início  $v_4$ , destino  $v_0$

Fila: -> $v_2$

Processados:  
 $v_4 \ v_1 \ v_3$

# Labirinto

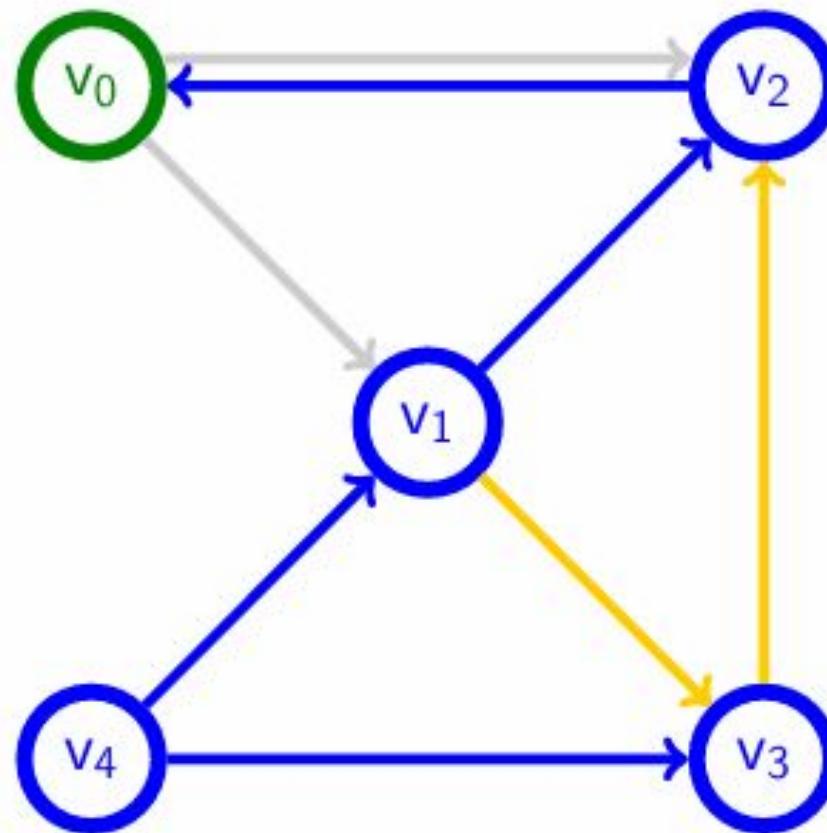


Início  $v_4$ , destino  $v_0$

Fila:

Processados:  
 $v_4 \ v_1 \ v_3 \ v_2$

# Labirinto

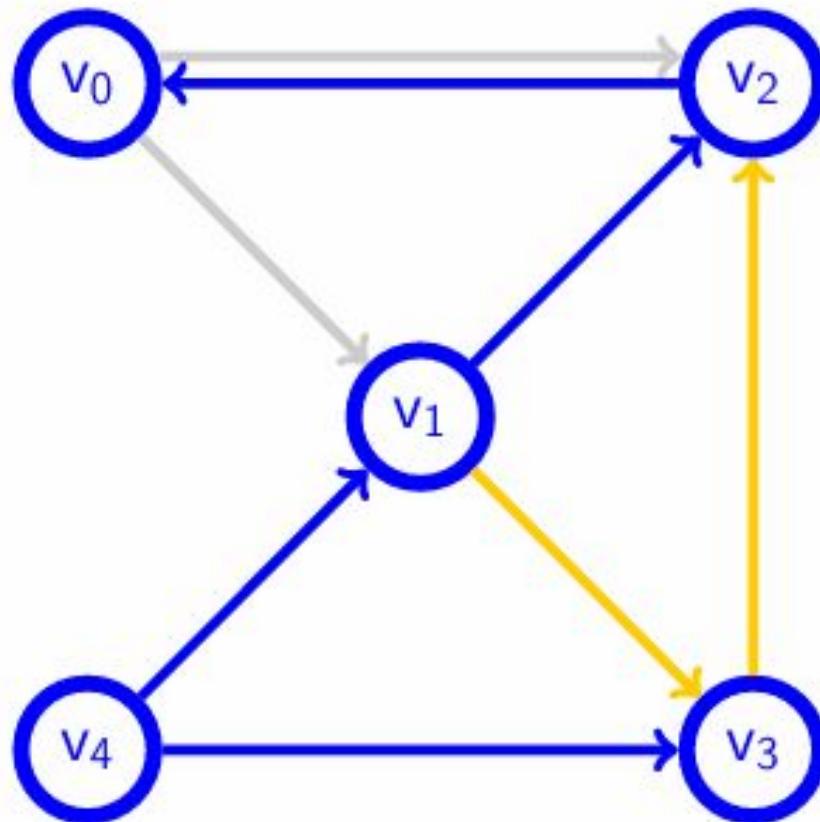


Início  $v_4$ , destino  $v_0$

Fila: -> $v_0$

Processados:  
 $v_4 \ v_1 \ v_3 \ v_2$

# Labirinto

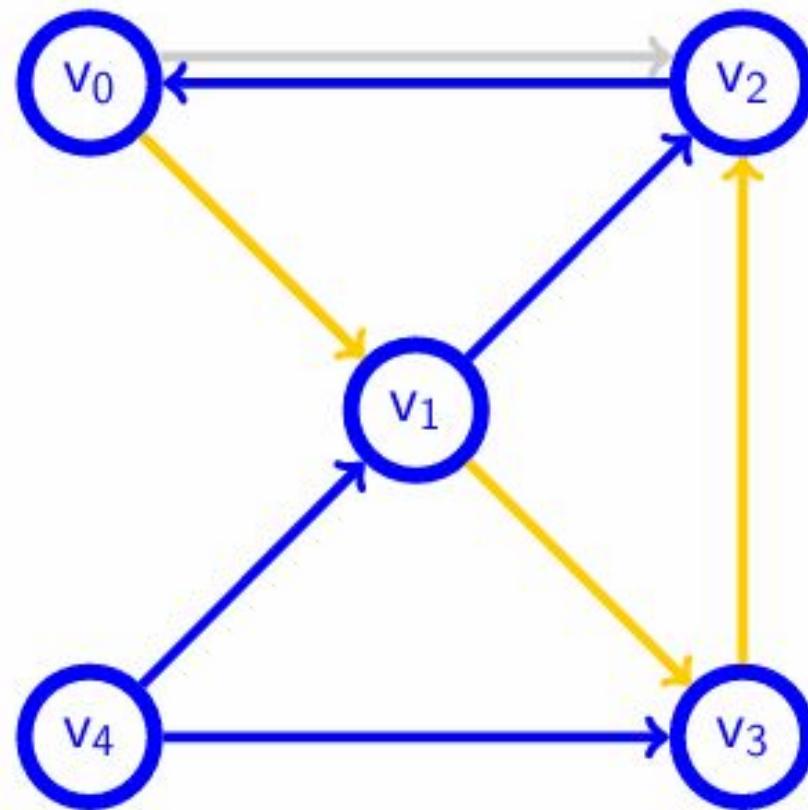


Início  $v_4$ , destino  $v_0$

Fila:

Processados:  
 $v_4 \ v_1 \ v_3 \ v_2 \ v_0$

# Labirinto



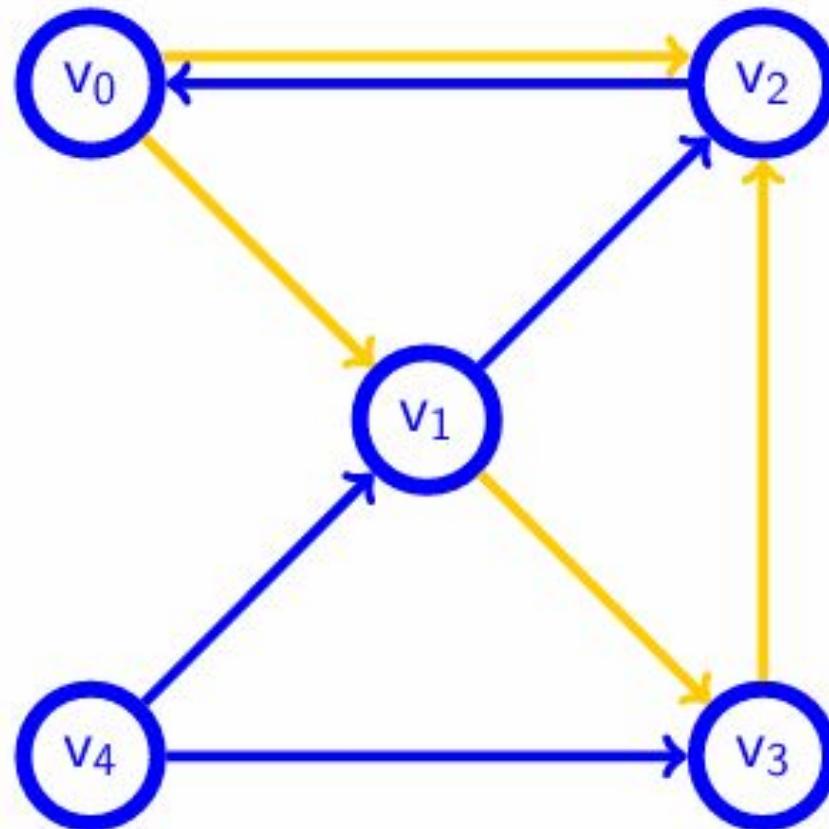
Início  $v_4$ , destino  $v_0$

Fila:

Processados:

$v_4 \ v_1 \ v_3 \ v_2 \ v_0$

# Labirinto



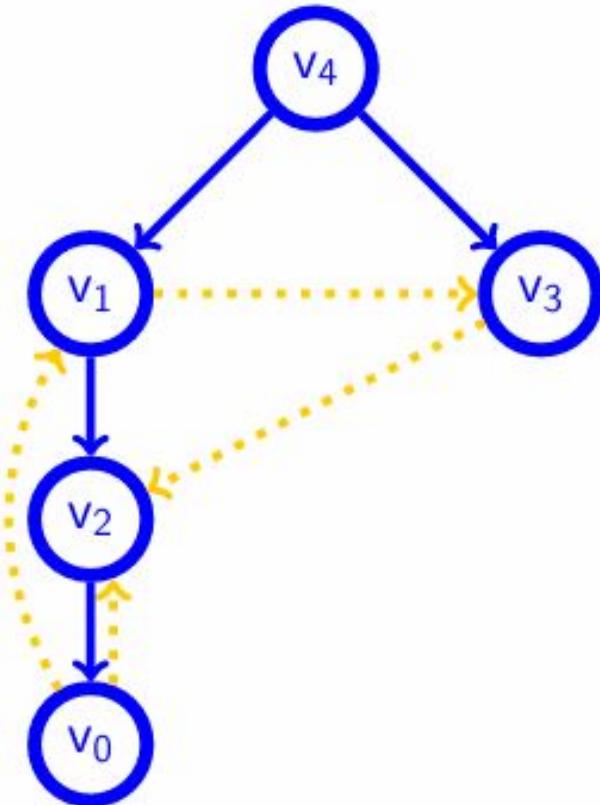
Início  $v_4$ , destino  $v_0$

Fila:

Processados:

$v_4 \ v_1 \ v_3 \ v_2 \ v_0$

# Labirinto



Resumo da Busca em Largura (iniciada em  $v_4$ )

Nó	Ant.	Dist.
0	2	3
1	4	1
2	1	2
3	4	1
4	-1	0

# Desafíos



# Desafio

- Implemente o algoritmo do labirinto.