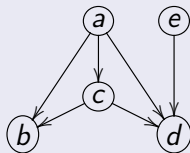


# Busca em profundidade

## Algoritmo

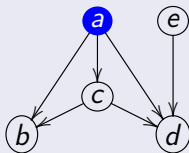
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se azul
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se vermelho



# Busca em profundidade

## Algoritmo

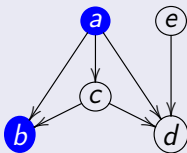
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

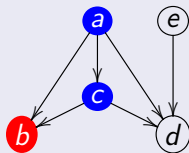
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

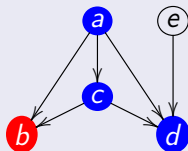
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

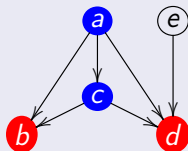
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

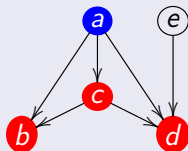
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

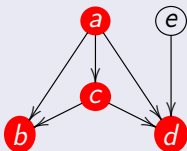
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se azul
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se vermelho

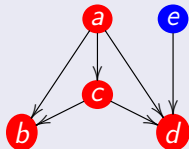




# Busca em profundidade

## Algoritmo

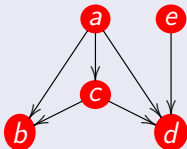
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Algoritmo

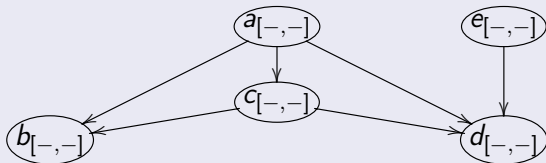
- Todos os vértices são inicializados com branco
- Quando um vértice é visitado pela primeira vez ele torna-se **azul**
- Quando sua lista de adjacentes foi totalmente explorada ele torna-se **vermelho**



# Busca em profundidade

## Tempos

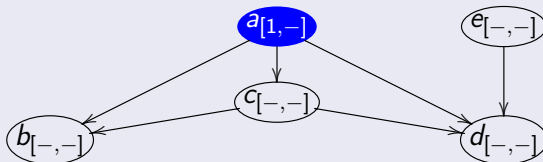
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

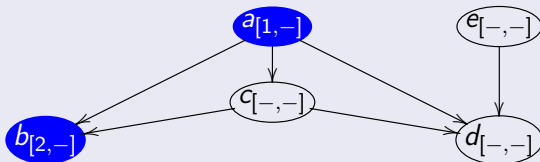
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

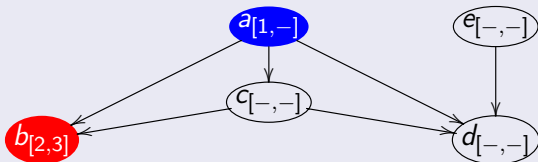
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

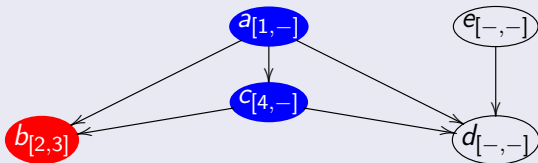
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

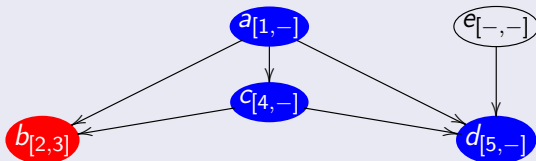
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo

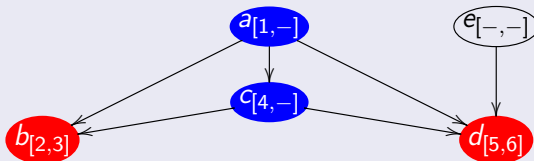




# Busca em profundidade

## Tempos

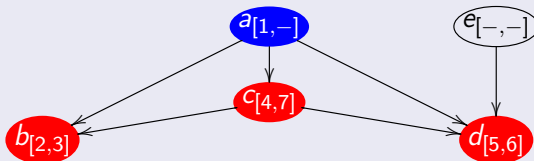
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

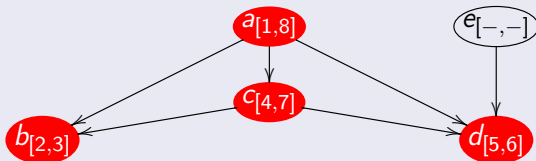
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

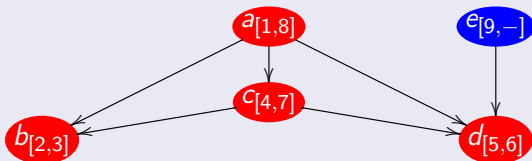
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

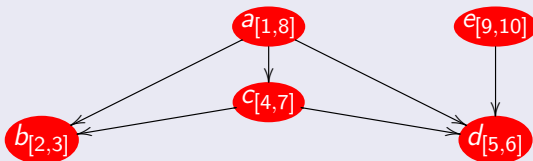
- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Tempos

- O tempo de descoberta  $d[v]$  é o momento em que o vértice  $v$  foi visitado pela primeira vez
- O tempo de término do exame da lista de adjacentes  $t[v]$  é o momento em que a visita a toda lista de vértices adjacentes a  $v$  foi concluída.
- $d[v]$  e  $t[v]$  são inteiros entre 1 e  $2V$ , onde  $V$  é o número de vértices do grafo



# Busca em profundidade

## Algoritmo - DFS

```
Require: Grafo  $G = (V, E)$   
for  $\forall u \in G$  do  
     $cor[u] = \text{branco}$   
     $pred[u] = -1$   
end for  
 $tempo = 0$   
for  $\forall u \in G$  do  
    if  $(cor[u] == \text{branco})$  then  
         $visita(u, tempo)$   
    end if  
end for
```

## Algoritmo - Visita

```
Require: Grafo  $G = (V, E)$   
Require: Vértice  $u$  e  $tempo$   
     $cor[u] = \text{azul}$   
     $tempo = tempo + 1$   
     $d[u] = tempo$   
    for  $\forall v \in Adj(u)$  do  
        if  $(cor[v] == \text{branco})$  then  
             $pred[v] = u$   
             $visita(v, tempo)$   
        end if  
    end for  
     $cor[u] = \text{vermelho}$   
     $tempo = tempo + 1$   
     $t[u] = tempo$ 
```

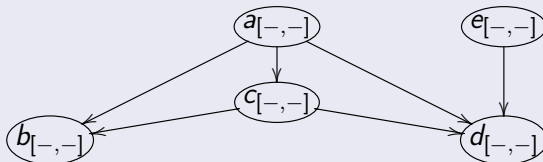
# Busca em profundidade

## Classificação das arestas

- De Árvore: uma Aresta  $(u,v)$  é de árvore se o vértice  $v$  foi visitado a primeira vez passando pela aresta  $(u,v)$
- De retorno: uma aresta  $(u,v)$  é uma aresta de retorno se esta conecta um vértice  $u$  com um predecessor  $v$  já presente em uma árvore de busca
- De avanço: não pertencem a árvore de busca em profundidade mas conectam um vértice a um descendente que pertence a árvore de busca
- De cruzamento: conectam vértice de uma mesma árvore de busca ou de árvores diferentes
- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento

# Busca em profundidade

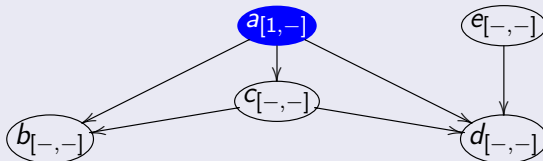
- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento





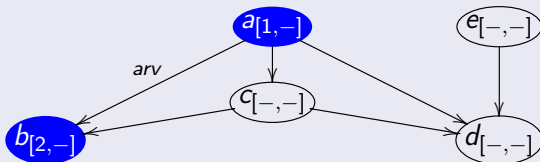
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



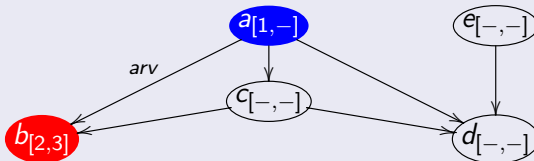
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



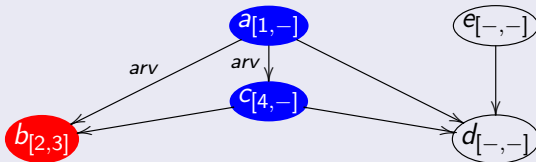
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



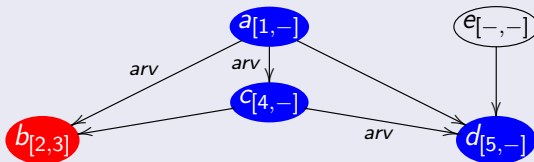
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



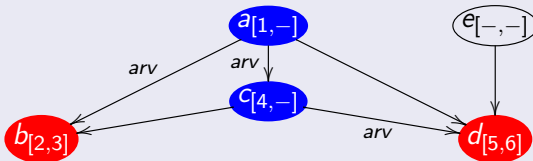
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



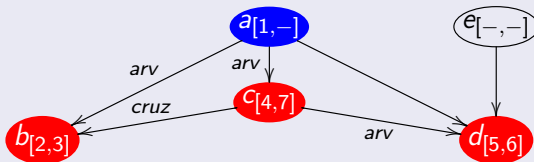
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



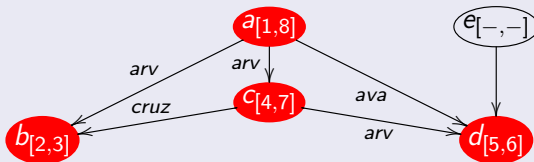
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



# Busca em profundidade

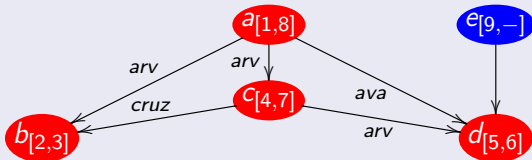
- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento





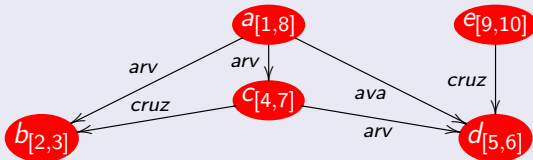
# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



# Busca em profundidade

- As arestas  $e = (u, v)$  podem ser classificadas pela cor do vértice  $v$  que é alcançado quando se passa pela aresta e primeira vez
  - Branco: aresta de árvore
  - Azul: aresta de retorno
  - Vermelha: (i) Se  $u$  é visitado antes de  $v$  então  $e$  é uma aresta de avanço; (ii) Se  $v$  é visitado antes de  $u$  então  $e$  é de cruzamento



# Busca em profundidade

## Teste de circuito

- Se uma aresta de retorno é encontrada na busca em profundidade então o grafo possui um ciclo
- Um grafo é acíclico se e somente se na busca em profundidade não for encontrada nenhuma aresta de retorno