Aula 5 - Propriedades e Aplicações de DFS

Luís Felipe

UFF

24 de Março de 2026

Luís Felipe

24/03/26

Busca em profundidade — Inicialização

Revisitando o DFS (visto em detalhes na última aula):

Luís Felipe

24/03/26

Busca em profundidade — Inicialização

Revisitando o DFS (visto em detalhes na última aula):

- $t \leftarrow 0$

  t é um contador compartilhado entre as chamadas recursivas

Luís Felipe

24/03/26

Busca em profundidade — Inicialização

Revisitando o DFS (visto em detalhes na última aula):

- $t \leftarrow 0$

  t é um contador compartilhado entre as chamadas recursivas
- Para todo vértice $v \in V(G)$ faça:
  - $PE(v) \leftarrow 0$

    $PE(v)$ é o tempo de entrada de v
  - $PS(v) \leftarrow 0$

    $PS(v)$ é o tempo de saída de v
  - $pai(v) \leftarrow null$

    ponteiros que definem a árvore de profundidade

Luís Felipe

24/03/26

Busca em profundidade — Inicialização

Revisitando o DFS (visto em detalhes na última aula):

- $t \leftarrow 0$
  t é um contador compartilhado entre as chamadas recursivas
- Para todo vértice $v \in V(G)$ faça:
  - $PE(v) \leftarrow 0$
    $PE(v)$ é o tempo de entrada de v
  - $PS(v) \leftarrow 0$
    $PS(v)$ é o tempo de saída de v
  - $pai(v) \leftarrow null$
    ponteiros que definem a árvore de profundidade
- Escolher um vértice qualquer $v \in V(G)$
  este vértice é chamado raiz da busca
- Executar $P(v)$

Luís Felipe

24/03/26

Busca em profundidade (DFS) — Procedimento

```
1  procedimento P(v, t)
2      t <- t + 1
3      PE(v) <- t
4
5      para todo vertice w em N(v) faca
6          se PE(w) = 0 entao
7              visitar aresta vw       // aresta azul (arvore)
8              pai(w) <- v
9              t <- P(w, t)
10         senao se PS(w) = 0 e w != pai(v) entao
11             visitar aresta vw       // aresta vermelha (retorno)
12
13     t <- t + 1
14     PS(v) <- t
15
16     retornar t
```

Luís Felipe

24/03/26

Busca em profundidade (DFS) — Procedimento

```
1  procedimento P(v, t)
2      t <- t + 1
3      PE(v) <- t
4
5      para todo vertice w em N(v) faca
6          se PE(w) = 0 entao
7              visitar aresta vw       // aresta azul (arvore)
8              pai(w) <- v
9              t <- P(w, t)
10         senao se PS(w) = 0 e w != pai(v) entao
11             visitar aresta vw       // aresta vermelha (retorno)
12
13     t <- t + 1
14     PS(v) <- t
15
16     retornar t
```

Intuição: Esgotar o filho antes de esgotar o pai.

Luís Felipe

24/03/26

Busca em profundidade (DFS) — Procedimento

```
1  procedimento P(v, t)
2      t <- t + 1
3      PE(v) <- t
4
5      para todo vertice w em N(v) faca
6          se PE(w) = 0 entao
7              visitar aresta vw        // aresta azul (arvore)
8              pai(w) <- v
9              t <- P(w, t)
10         senao se PS(w) = 0 e w != pai(v) entao
11             visitar aresta vw        // aresta vermelha (retorno)
12
13     t <- t + 1
14     PS(v) <- t
15
16     retornar t
```

Intuição: Esgotar o filho antes de esgotar o pai. A próxima aresta a ser visitada parte sempre do vértice mais recente na busca.

Luís Felipe

24/03/26

Busca em profundidade (DFS) — Procedimento

```
1  procedimento P(v, t)
2      t <- t + 1
3      PE(v) <- t
4
5      para todo vertice w em N(v) faca
6          se PE(w) = 0 entao
7              visitar aresta vw        // aresta azul (arvore)
8              pai(w) <- v
9              t <- P(w, t)
10         senao se PS(w) = 0 e w != pai(v) entao
11             visitar aresta vw        // aresta vermelha (retorno)
12
13     t <- t + 1
14     PS(v) <- t
15
16     retornar t
```

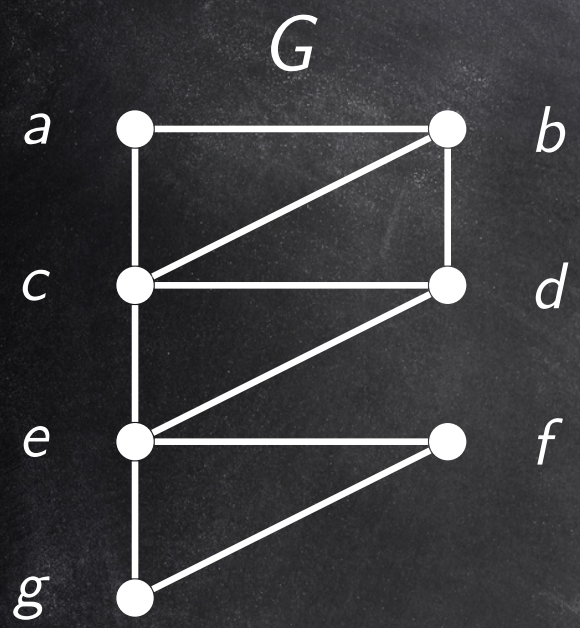
Intuição: Esgotar o filho antes de esgotar o pai. A próxima aresta a ser visitada parte sempre do vértice mais recente na busca.

Complexidade: $O(n + m)$ (provado na aula passada).

Luís Felipe

24/03/26

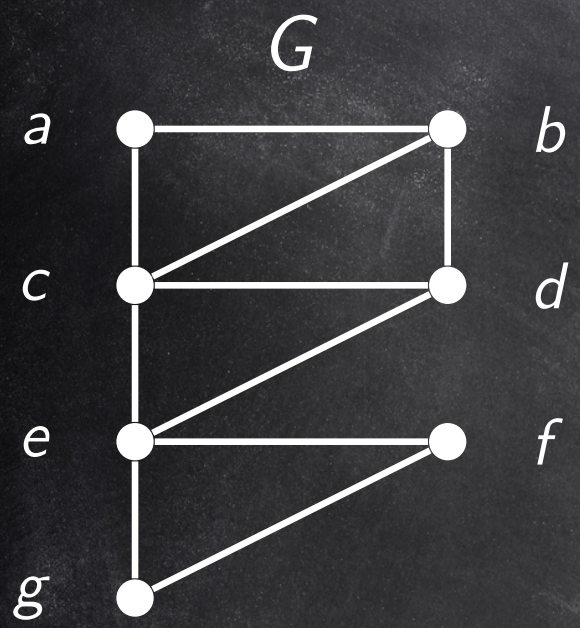
Execução da DFS



Luís Felipe

24/03/26

Execução da DFS

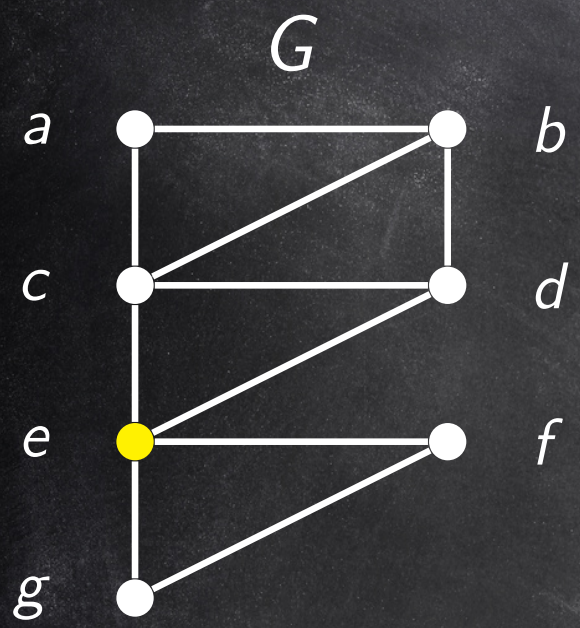


v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	0	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS

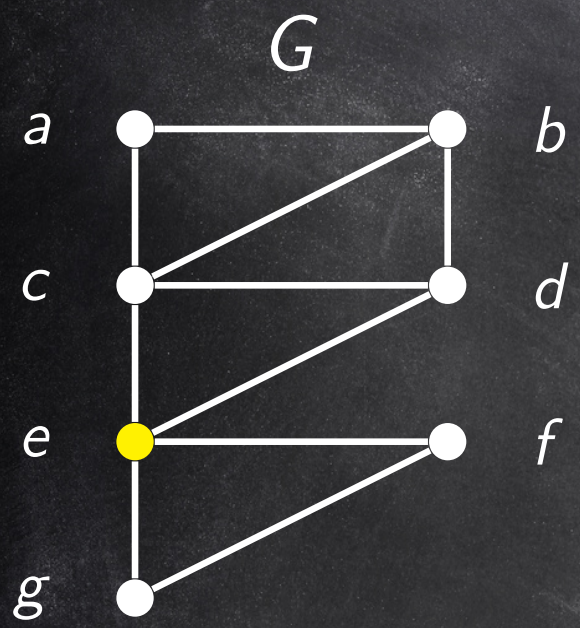


v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	0	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS



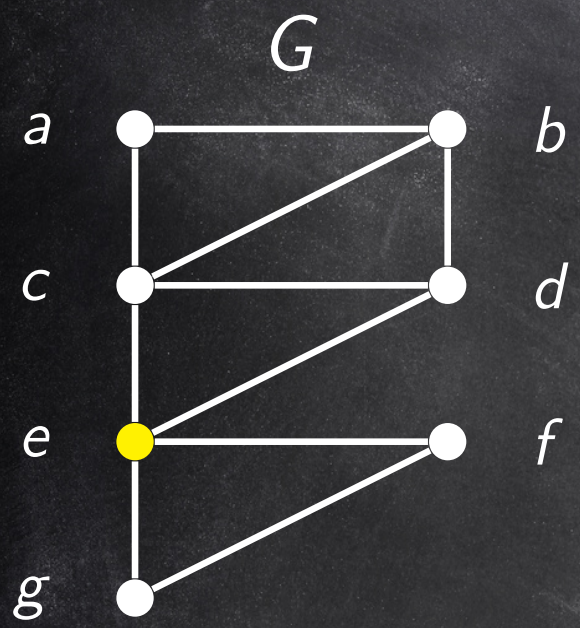
$P(e)$

v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	0	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS



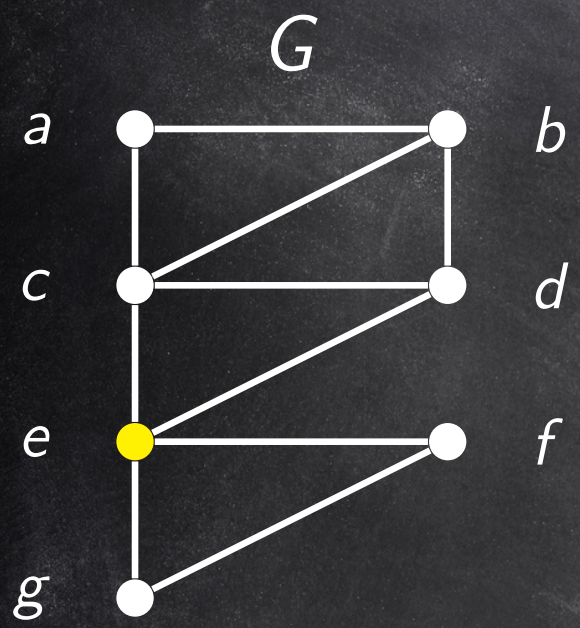
$P(e)$

v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS



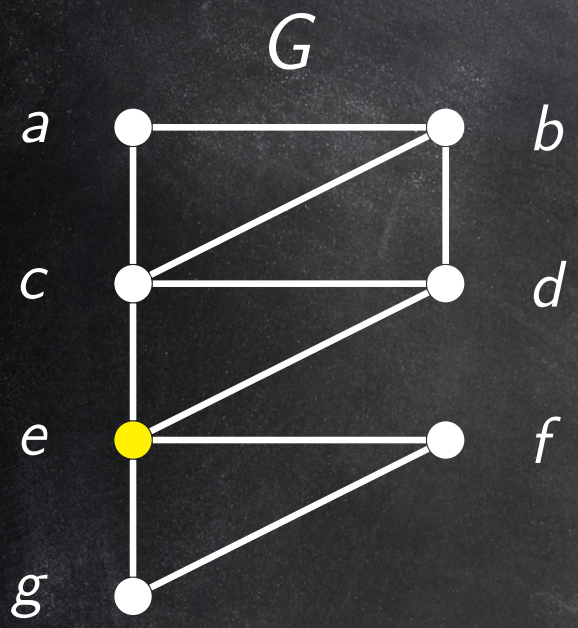
$P(e)$

v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS



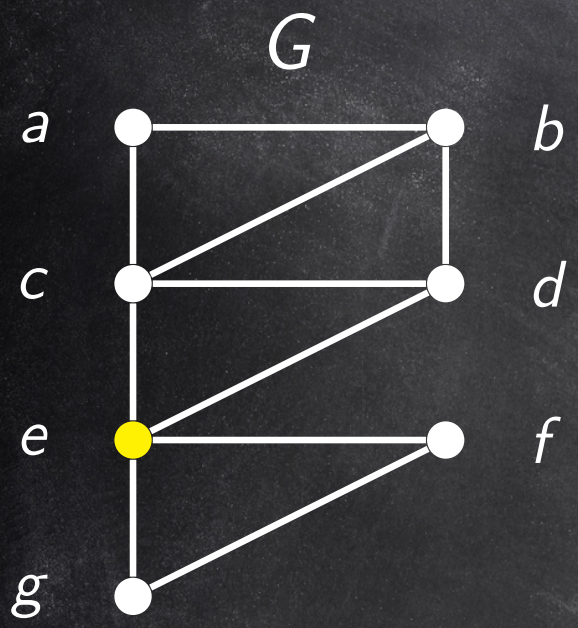
$P(e) \rightarrow N(e) = \{c, d, f, g\}$

v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS



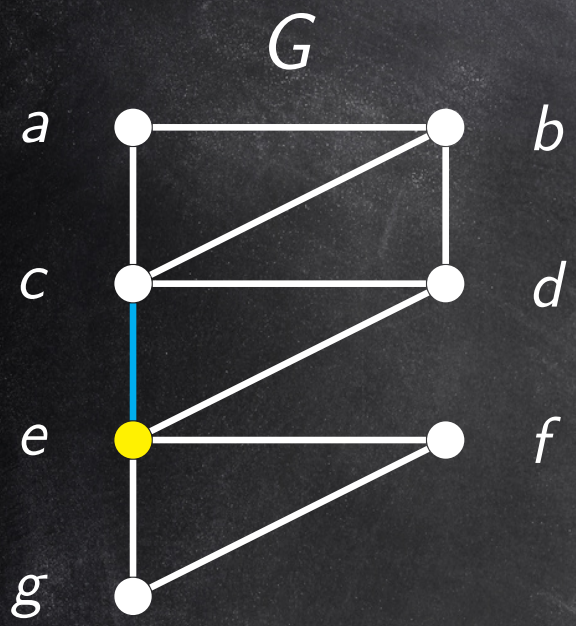
$P(e) \rightarrow N(e) = \{c, d, f, g\}$

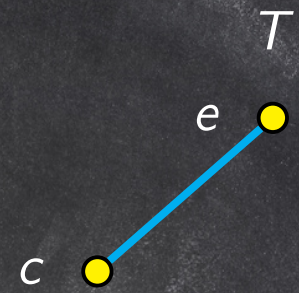
v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





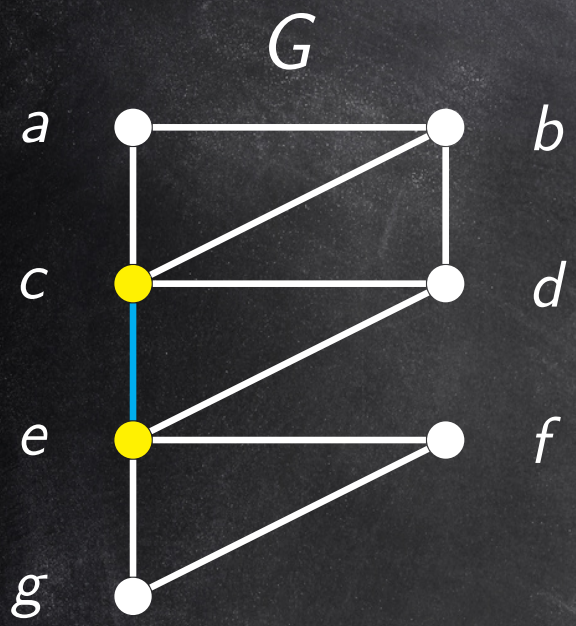
$P(e) \rightarrow N(e) = \{c, d, f, g\}$

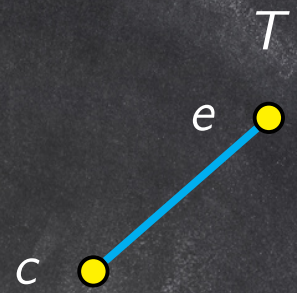
v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

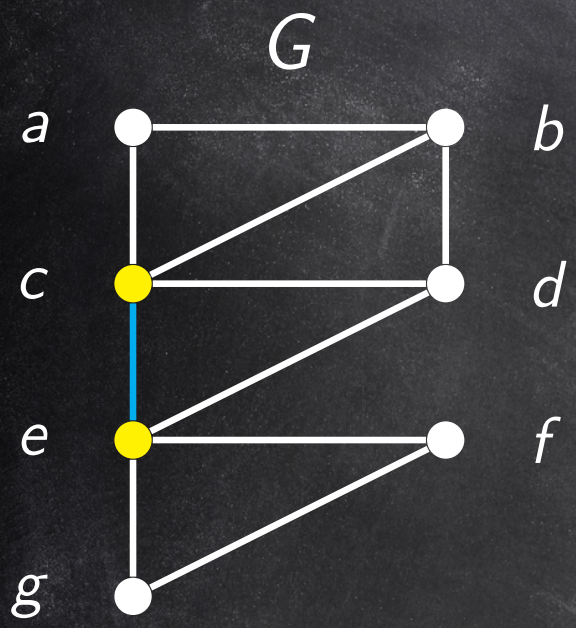
$P(c)$

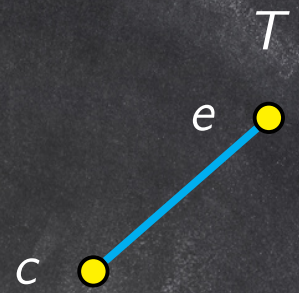
v	a	b	c	d	e	f	g
PE(v)	0	0	0	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

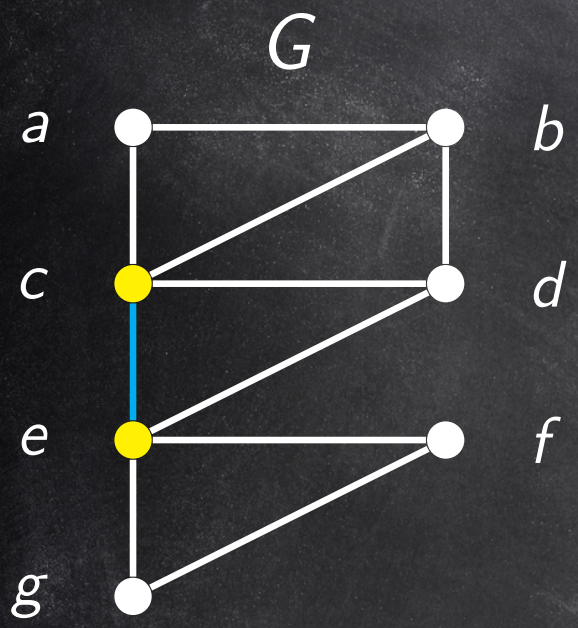
$P(c)$

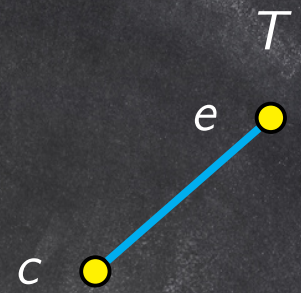
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

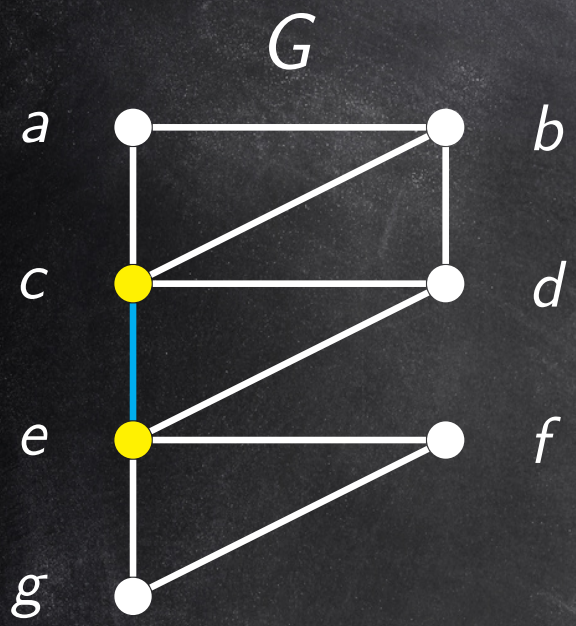
$P(c) \rightarrow N(c) = \{a, b, d, e\}$

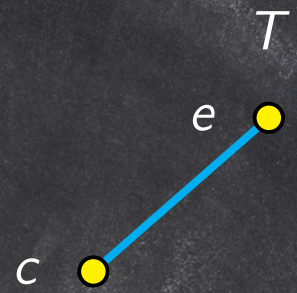
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

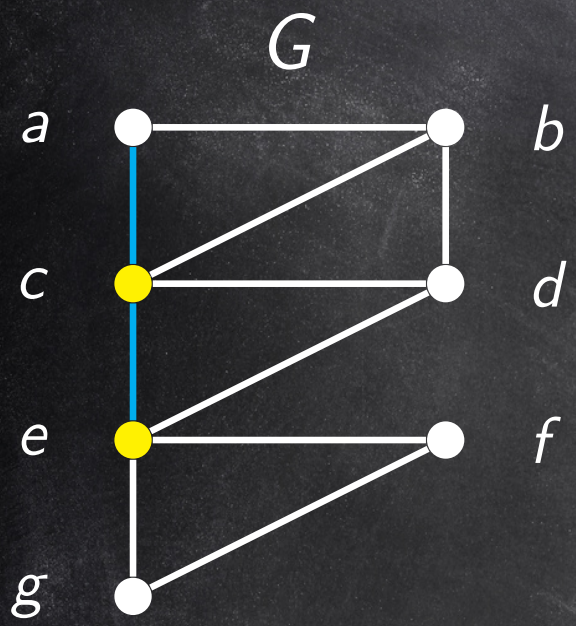
$P(c) \rightarrow N(c) = \{a, b, d, e\}$

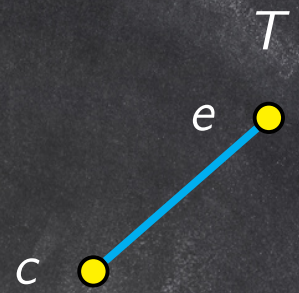
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

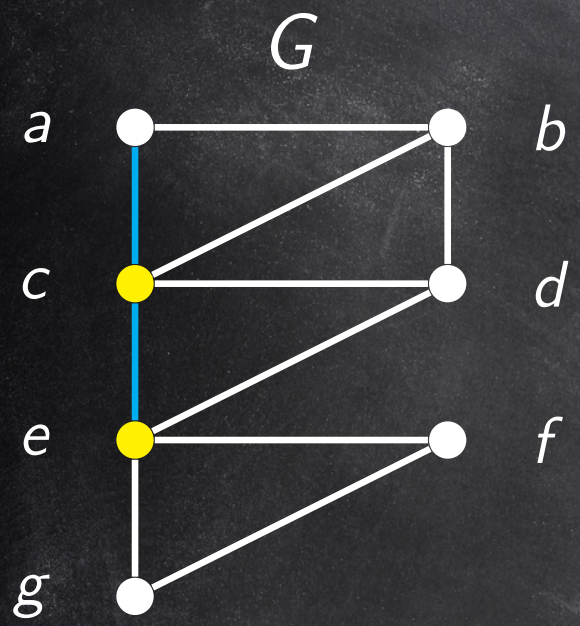
$P(c) \rightarrow N(c) = \{a, b, d, e\}$

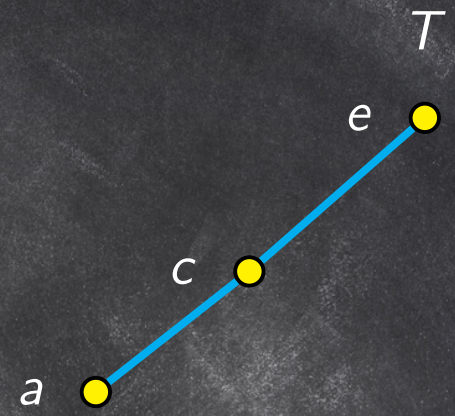
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

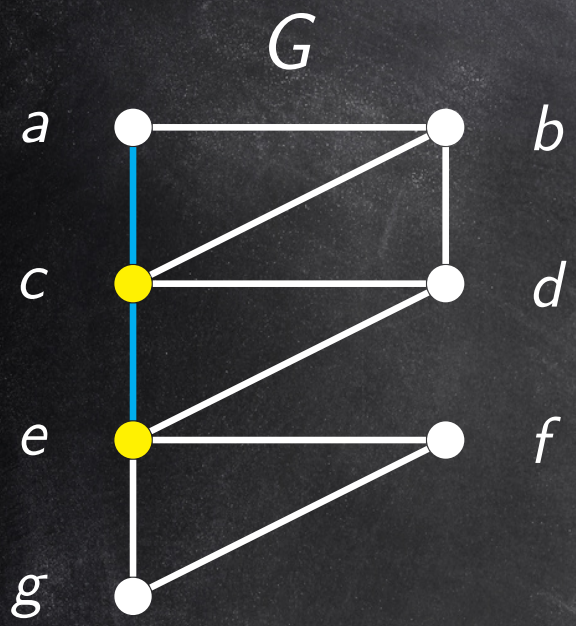
$P(c) \rightarrow N(c) = \{a, b, d, e\}$

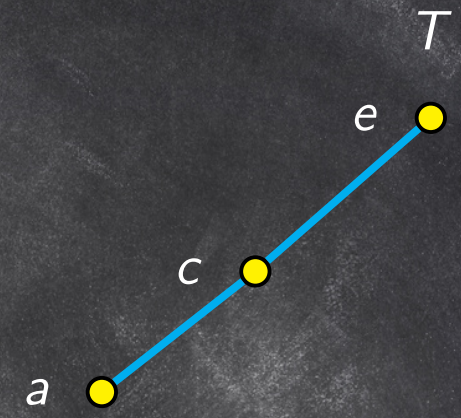
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

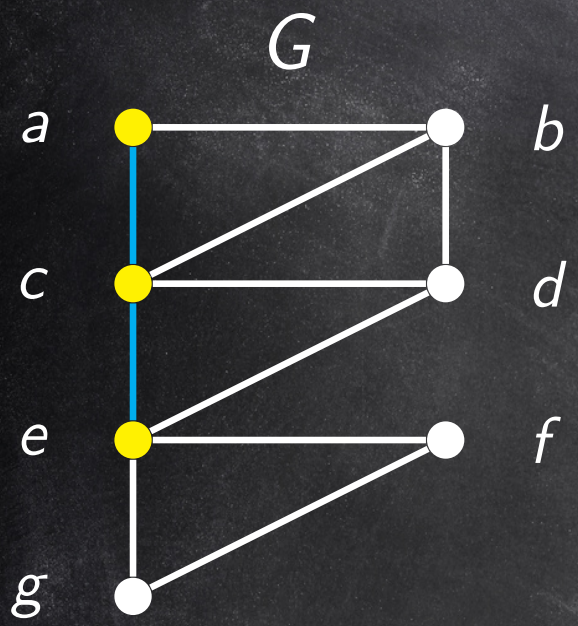
$P(a)$

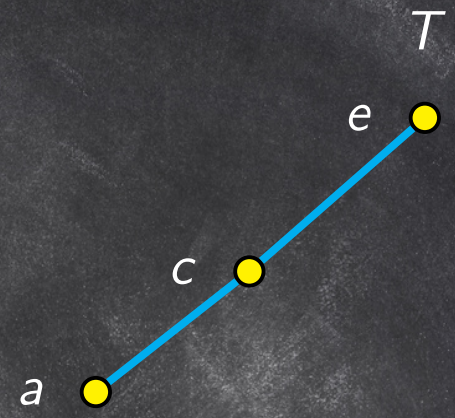
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

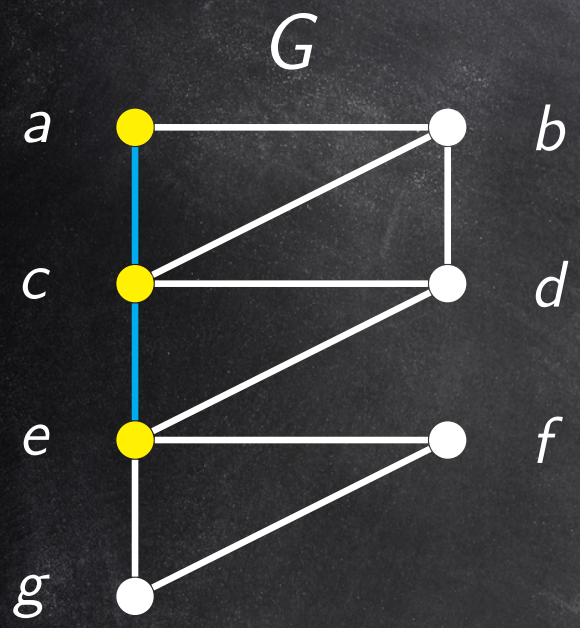
$P(a)$

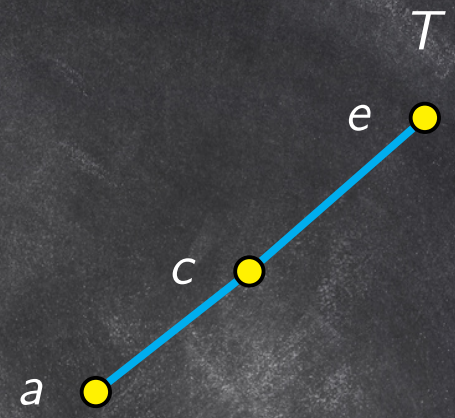
v	a	b	c	d	e	f	g
PE(v)	0	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

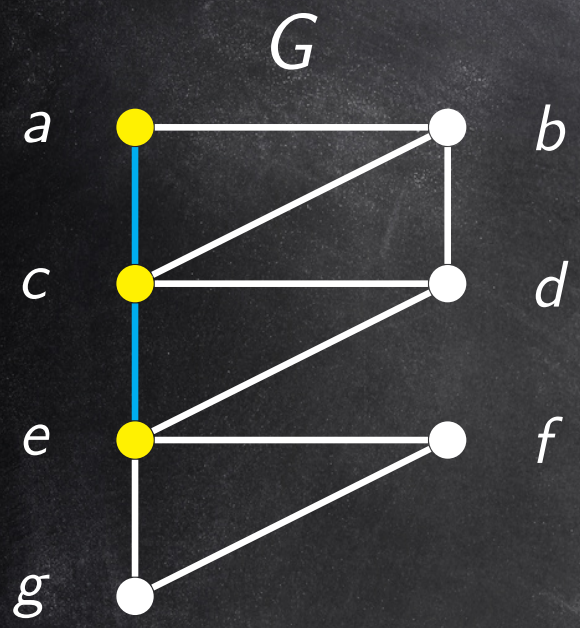
$P(a)$

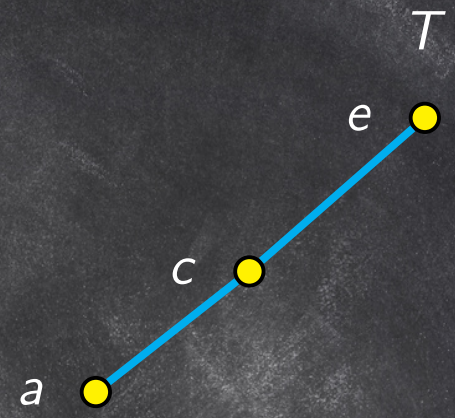
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

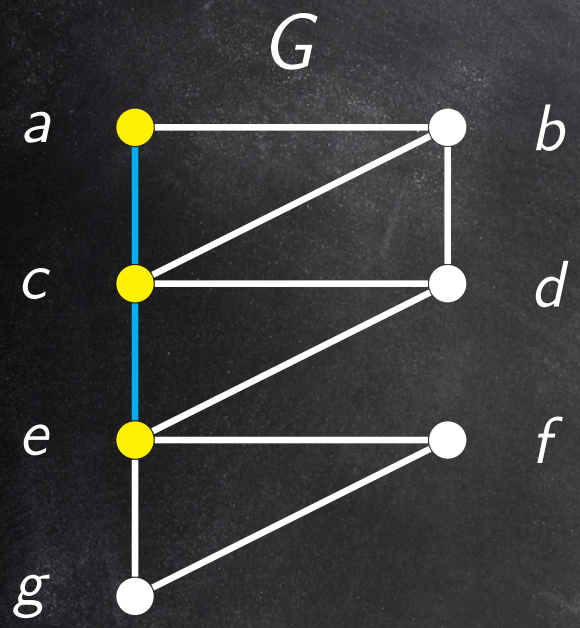
$P(a) \rightarrow N(a) = \{b, c\}$

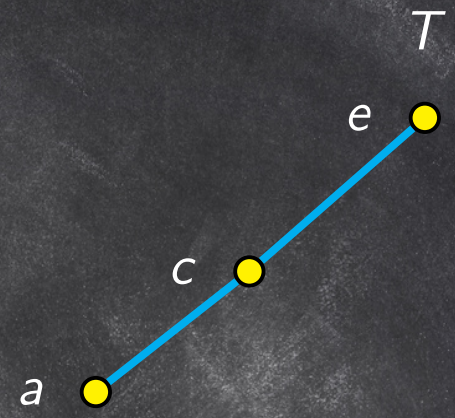
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

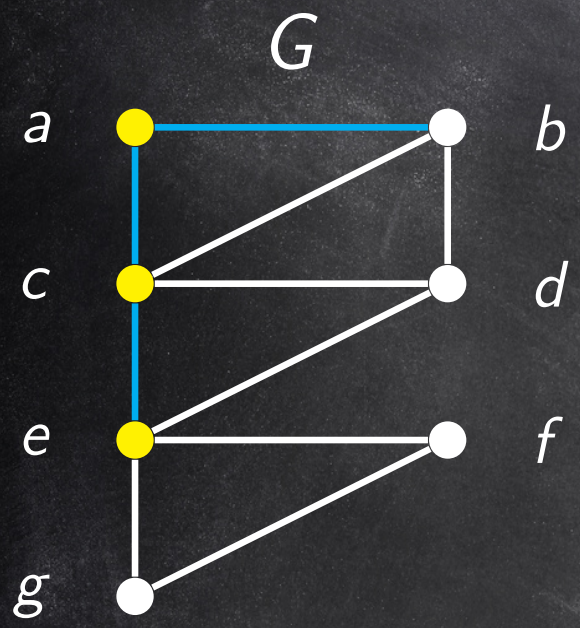
$P(a) \rightarrow N(a) = \{b, c\}$

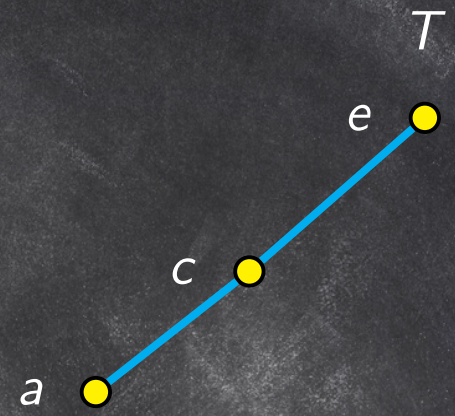
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

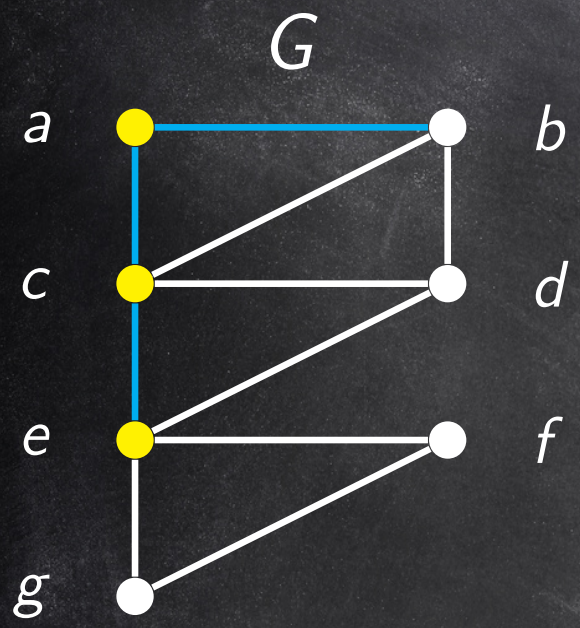
$P(a) \rightarrow N(a) = \{b, c\}$

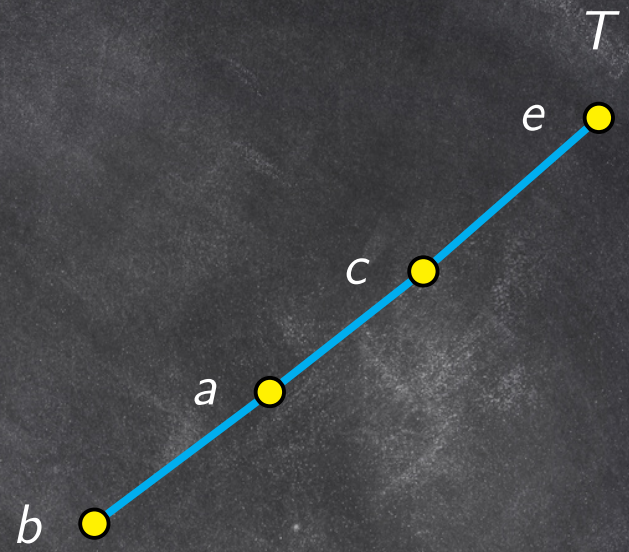
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

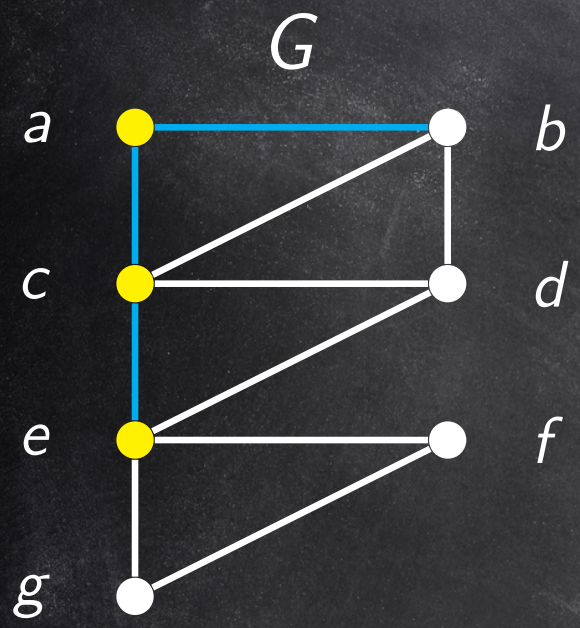
$P(a) \rightarrow N(a) = \{b, c\}$

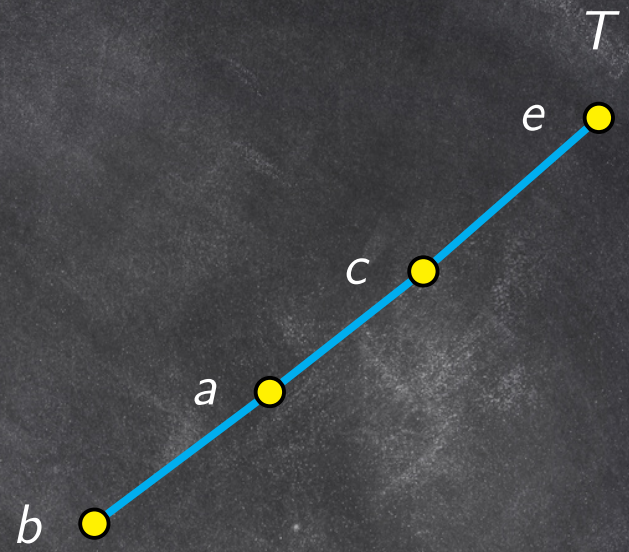
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

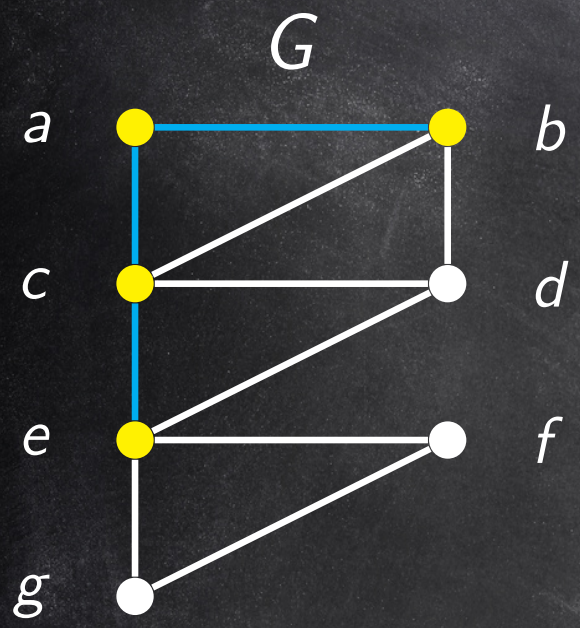
$P(b)$

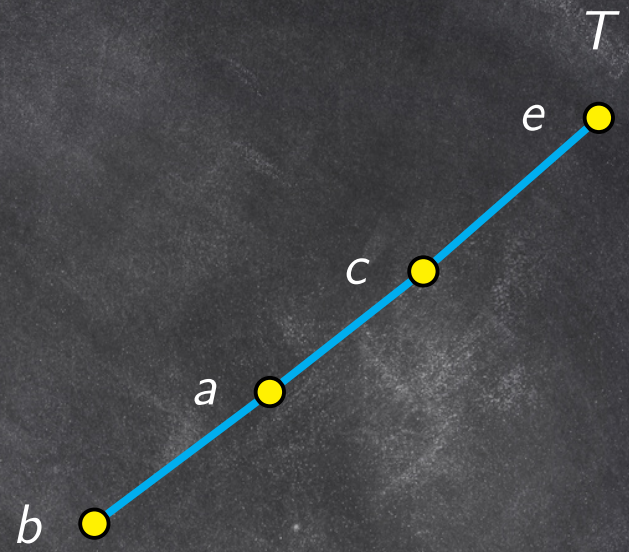
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

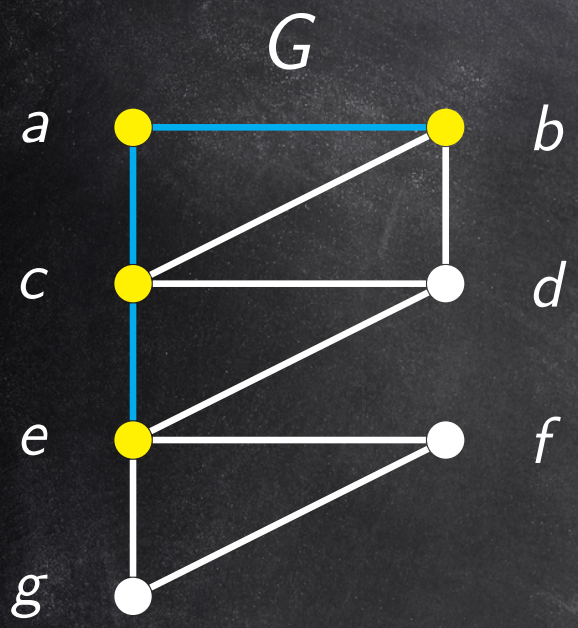
$P(b)$

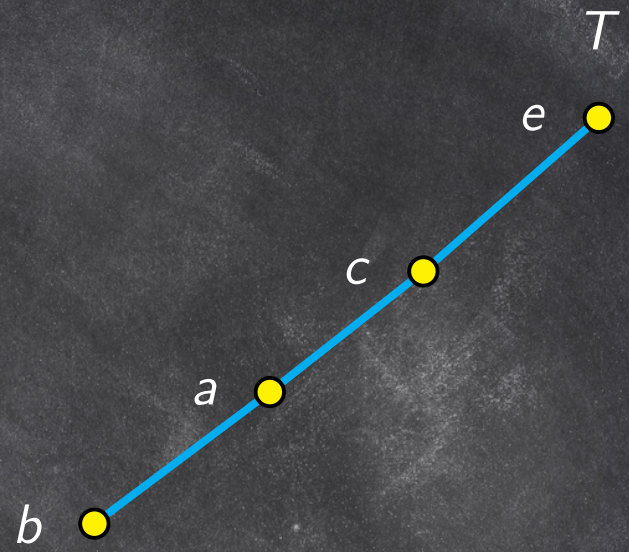
v	a	b	c	d	e	f	g
PE(v)	3	0	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

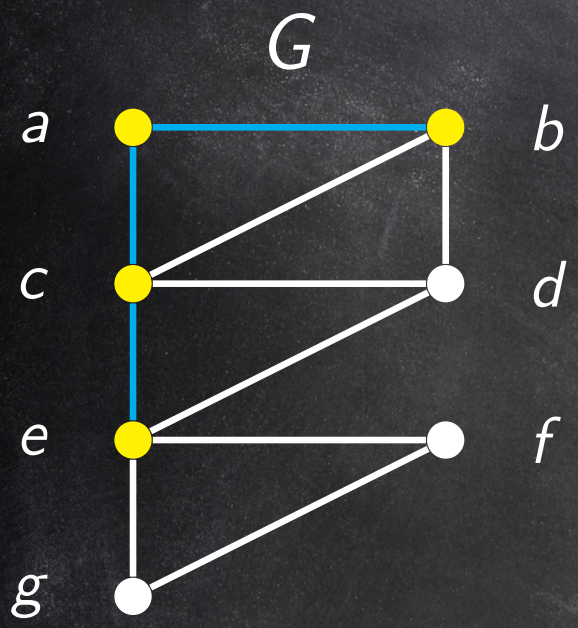
$P(b)$

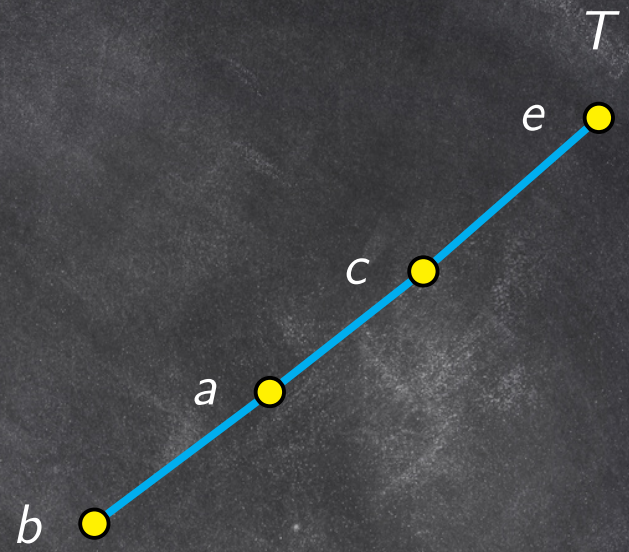
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

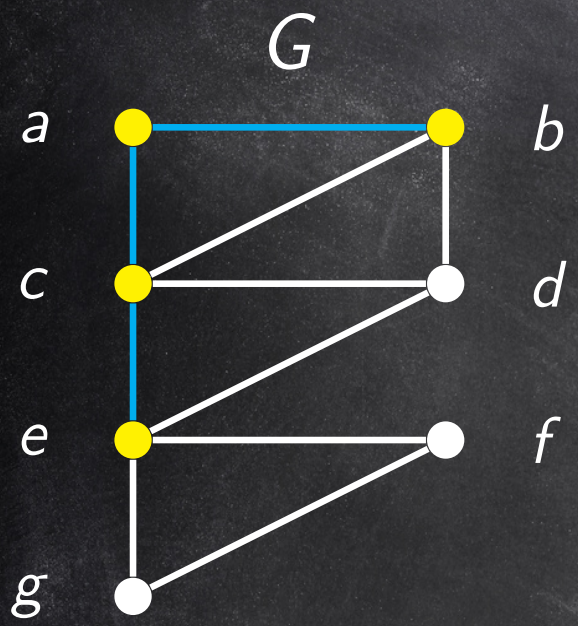
$P(b) \rightarrow N(b) = \{a, c, d\}$

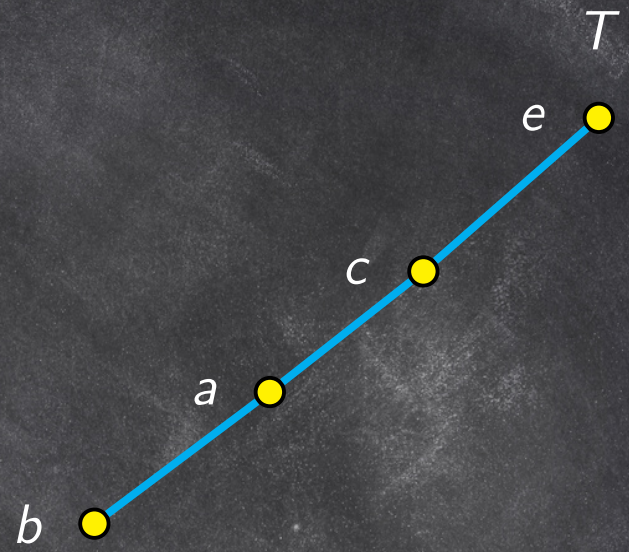
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

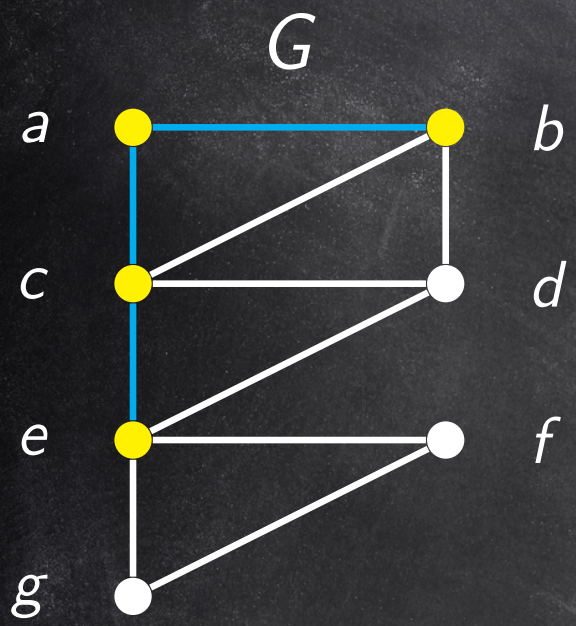
$P(b) \rightarrow N(b) = \{a, c, d\}$

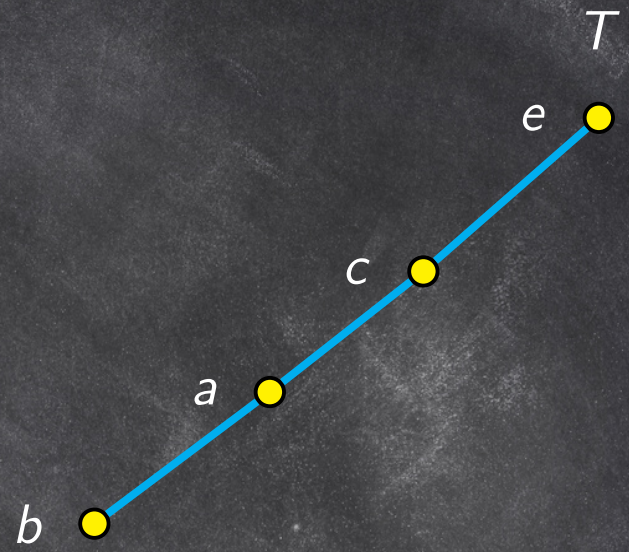
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

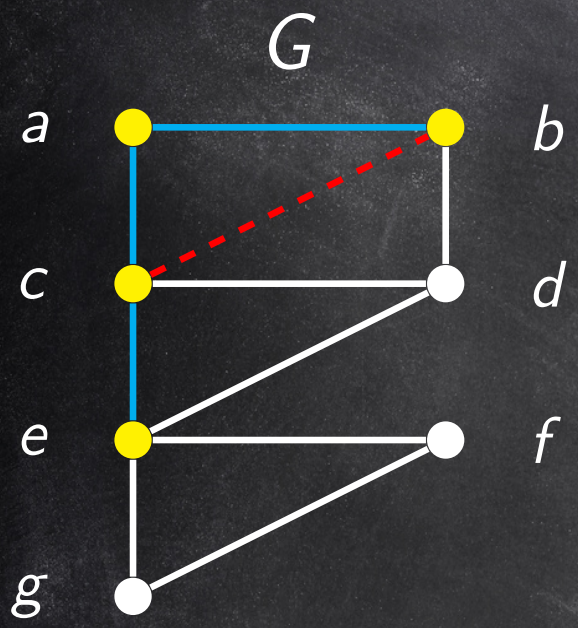
$P(b) \rightarrow N(b) = \{a, c, d\}$

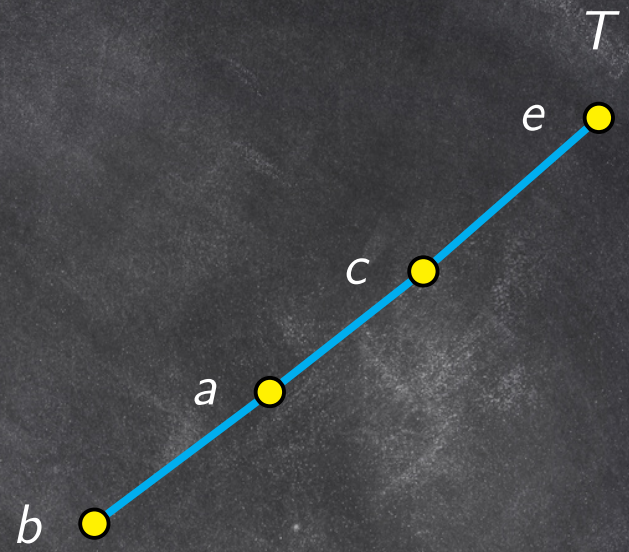
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

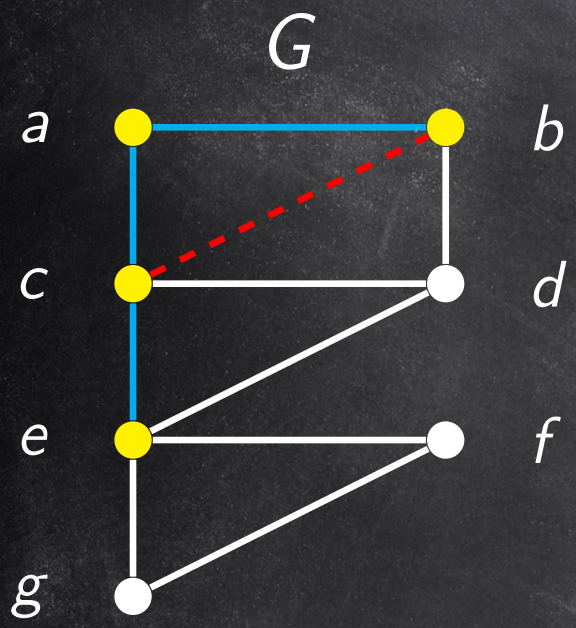
$P(b) \rightarrow N(b) = \{a, c, d\}$

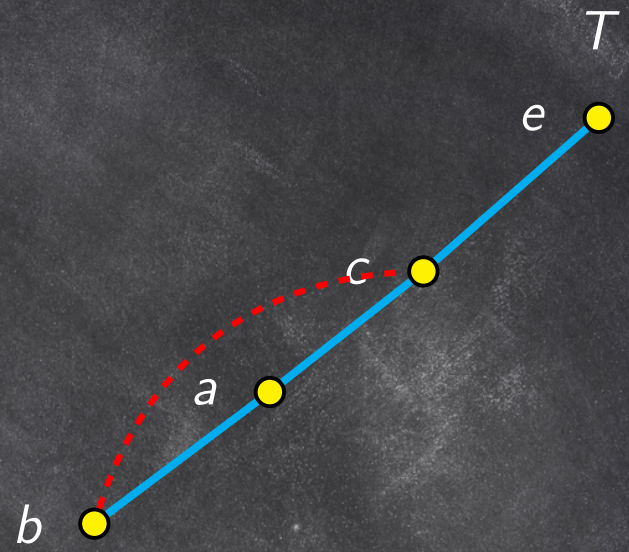
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

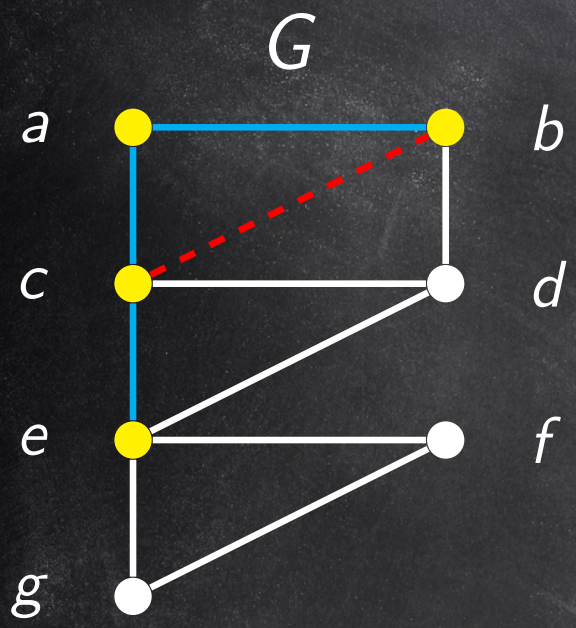
$P(b) \rightarrow N(b) = \{a, c, d\}$

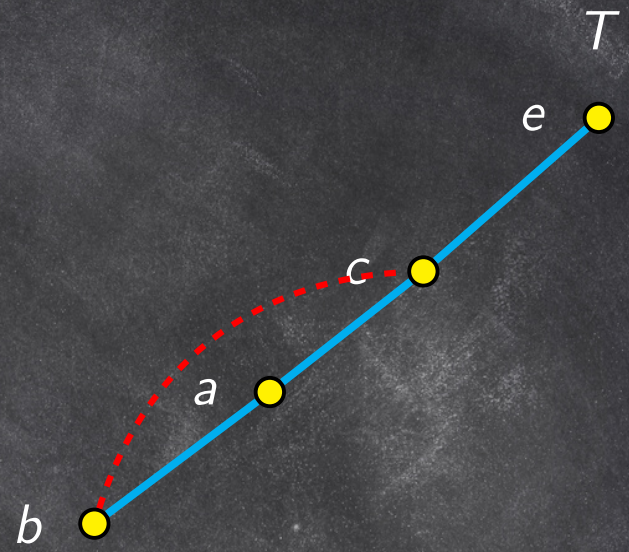
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

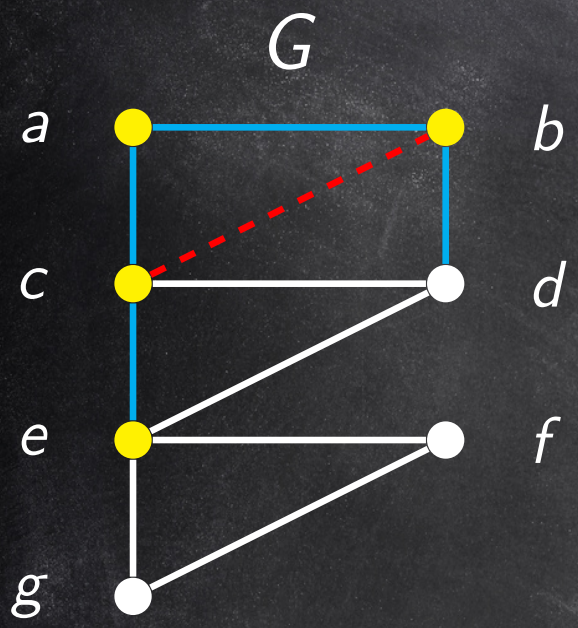
$P(b) \rightarrow N(b) = \{a, c, d\}$

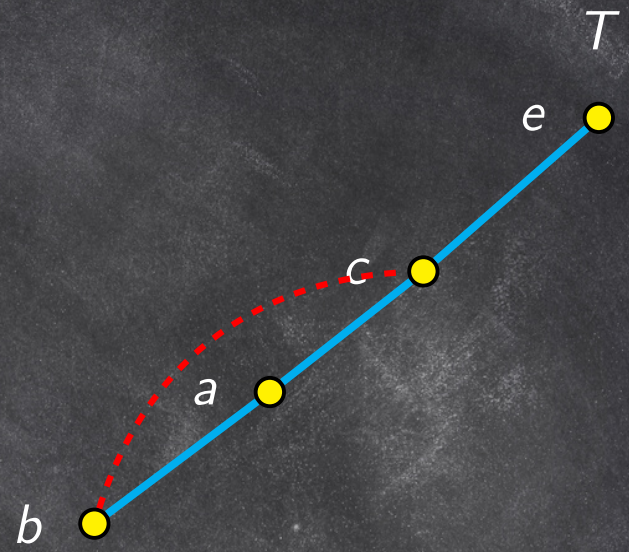
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

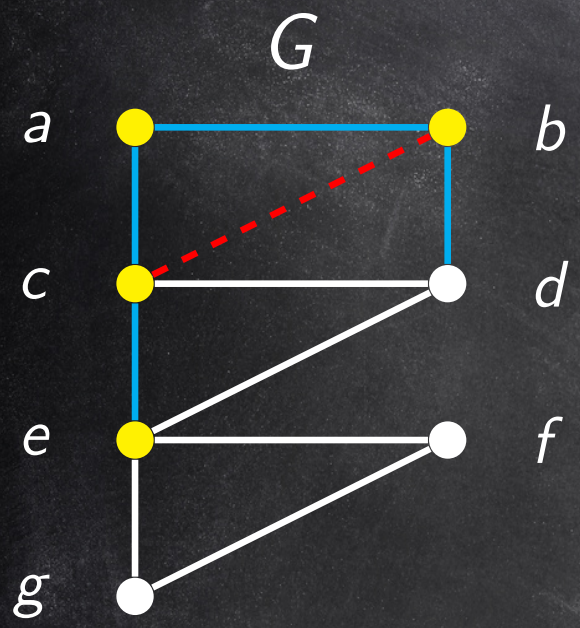
$P(b) \rightarrow N(b) = \{a, c, d\}$

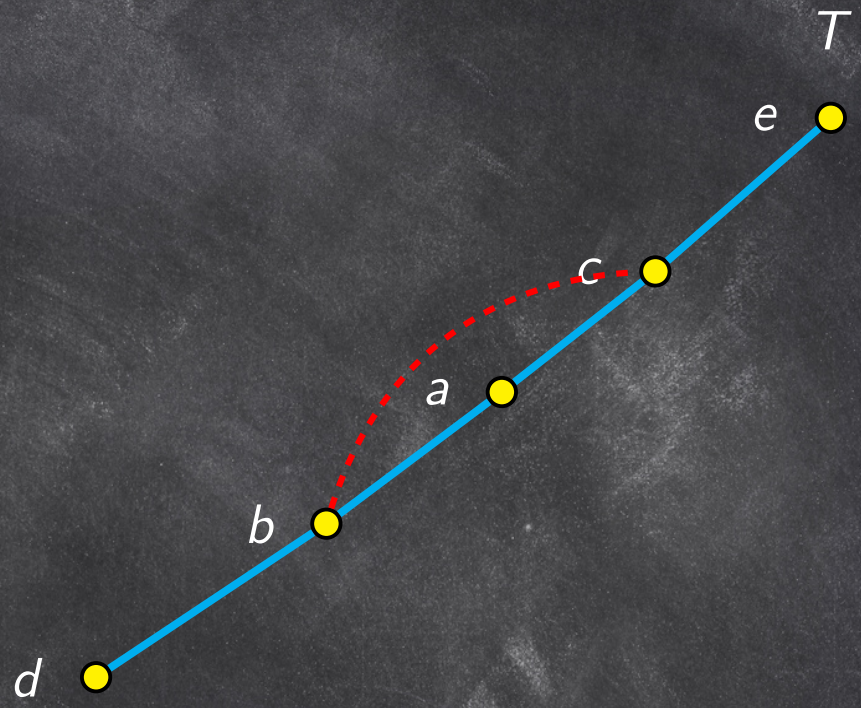
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

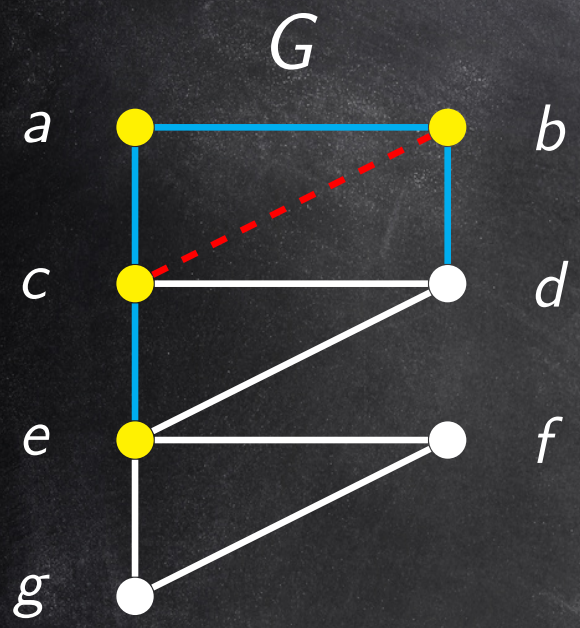
$P(b) \rightarrow N(b) = \{a, c, d\}$

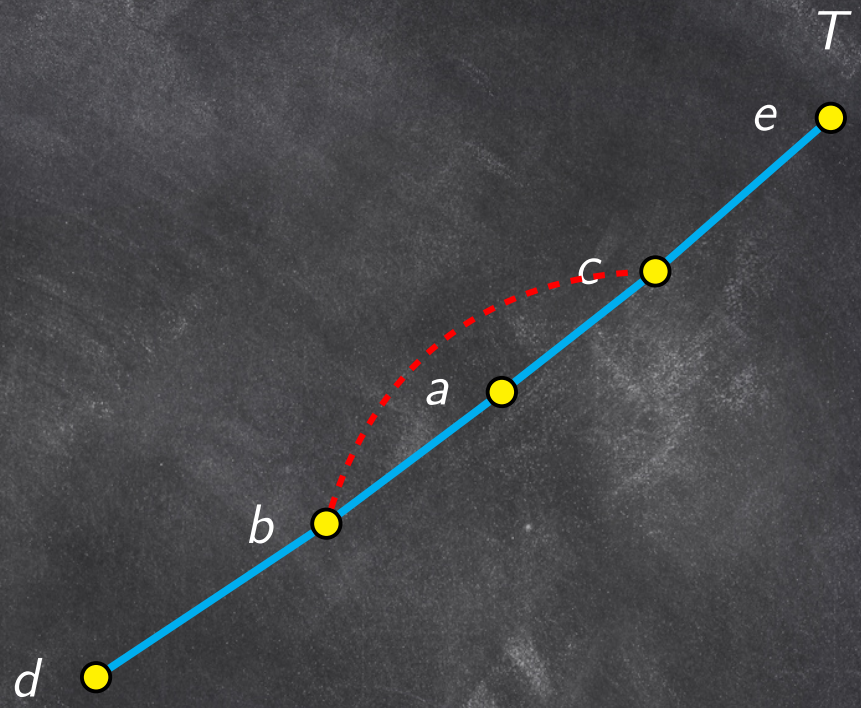
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

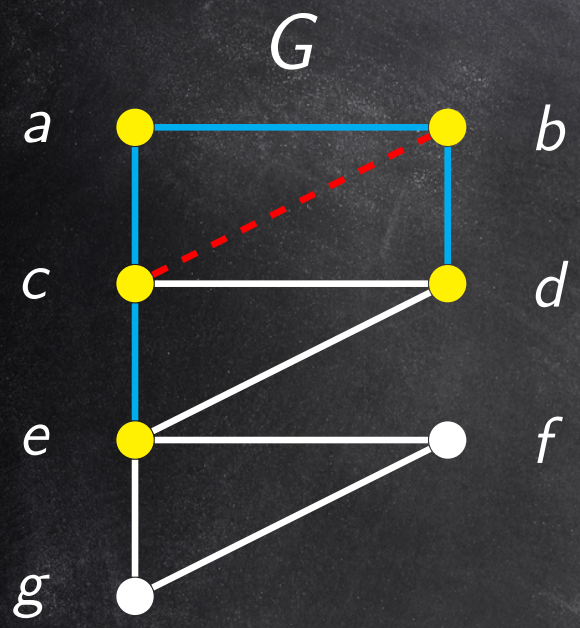
$P(d)$

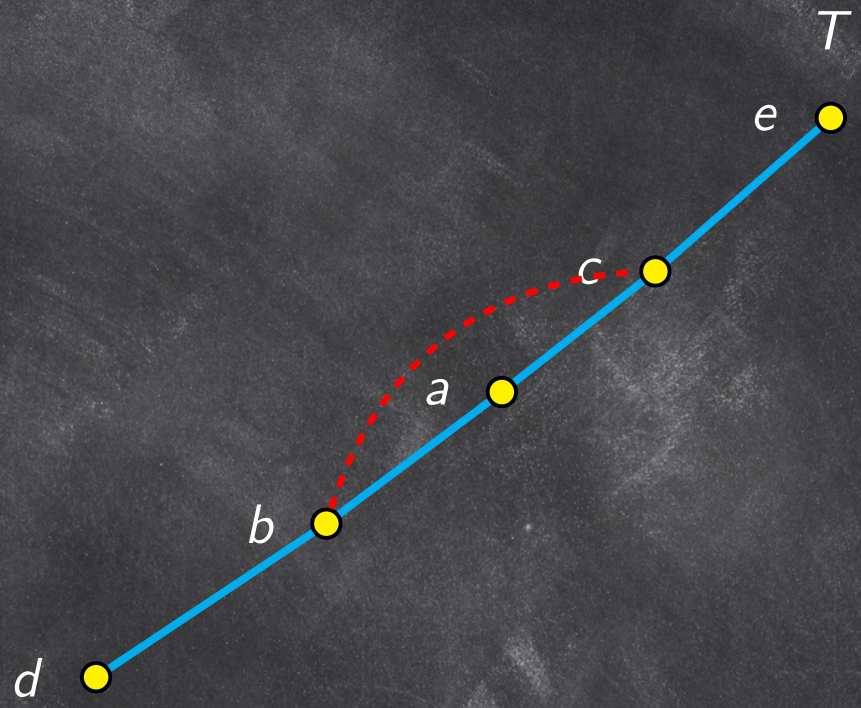
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

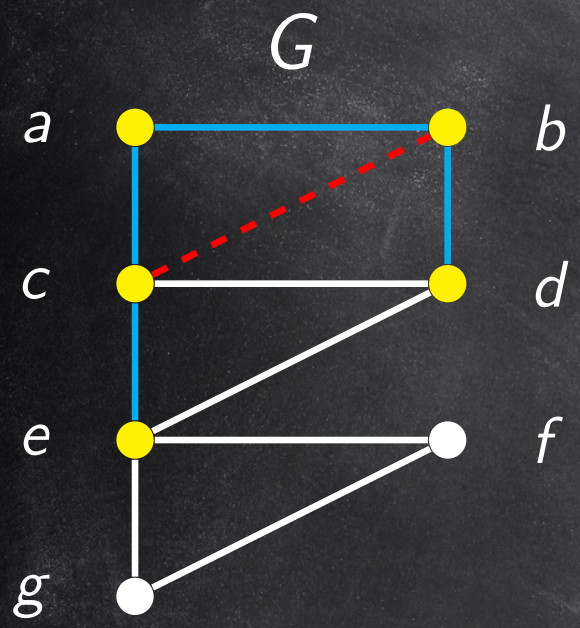
$P(d)$

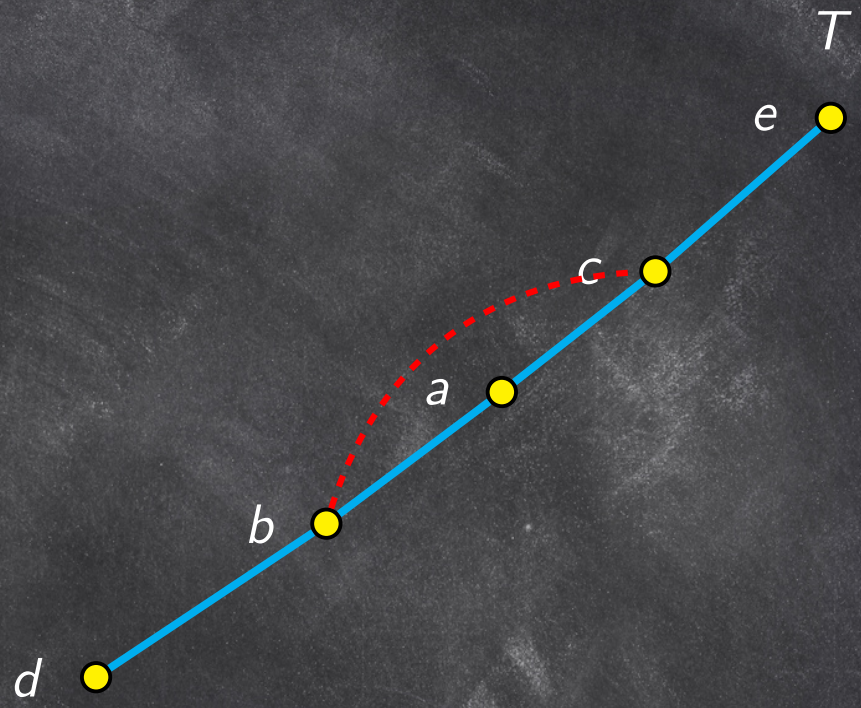
v	a	b	c	d	e	f	g
PE(v)	3	4	2	0	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

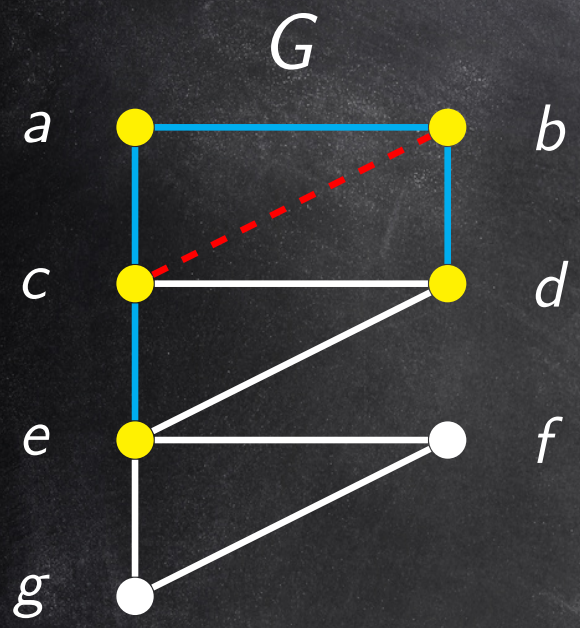
$P(d)$

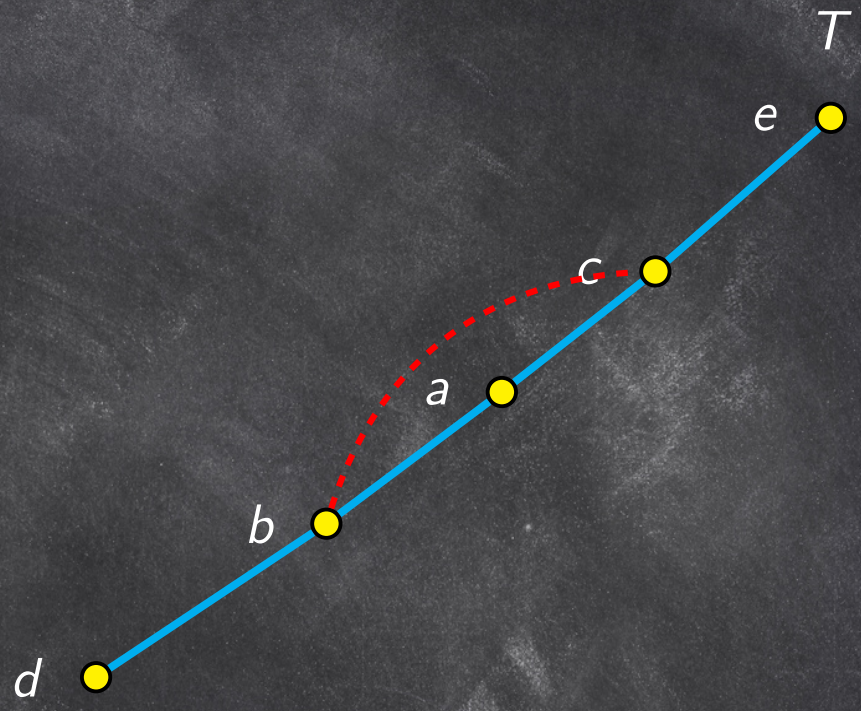
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

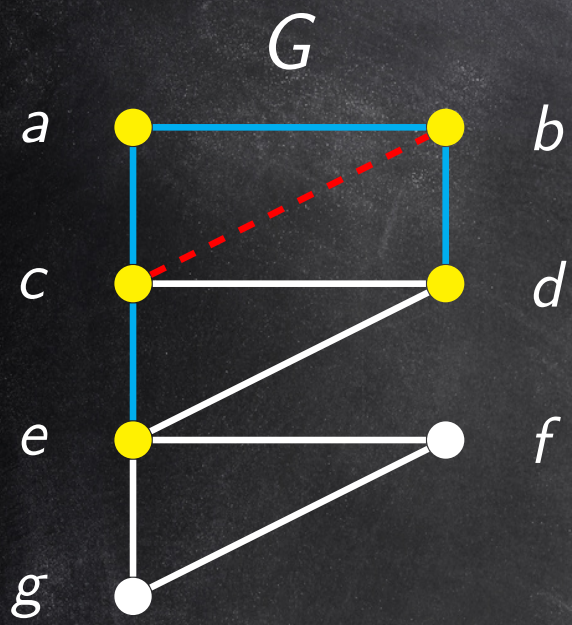
$P(d) \rightarrow N(d) = \{b, c, e\}$

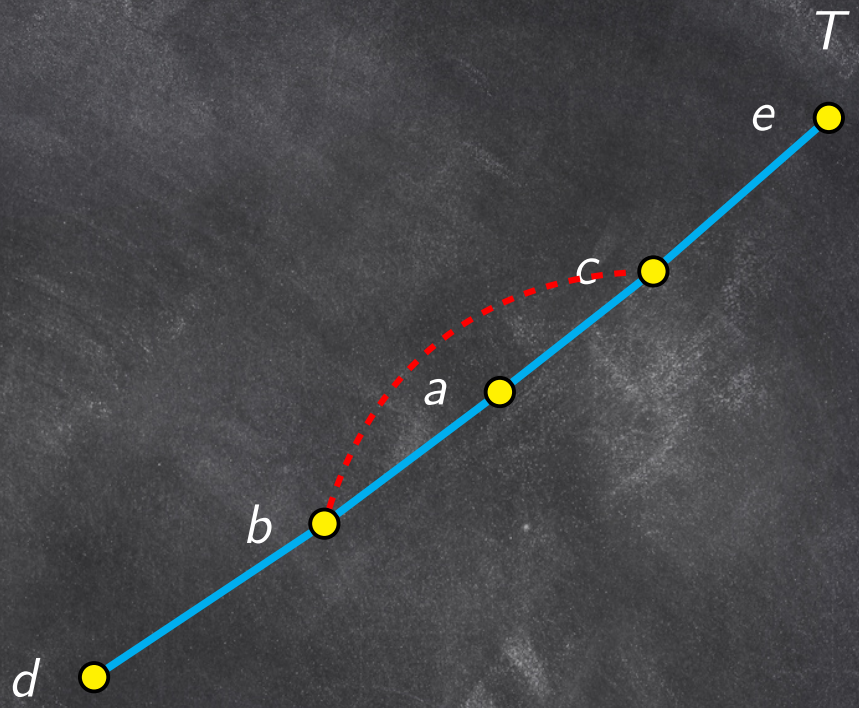
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

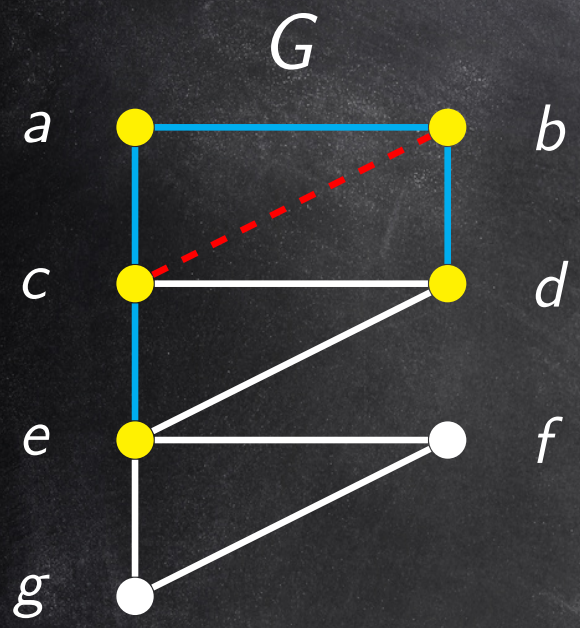
$P(d) \rightarrow N(d) = \{b, c, e\}$

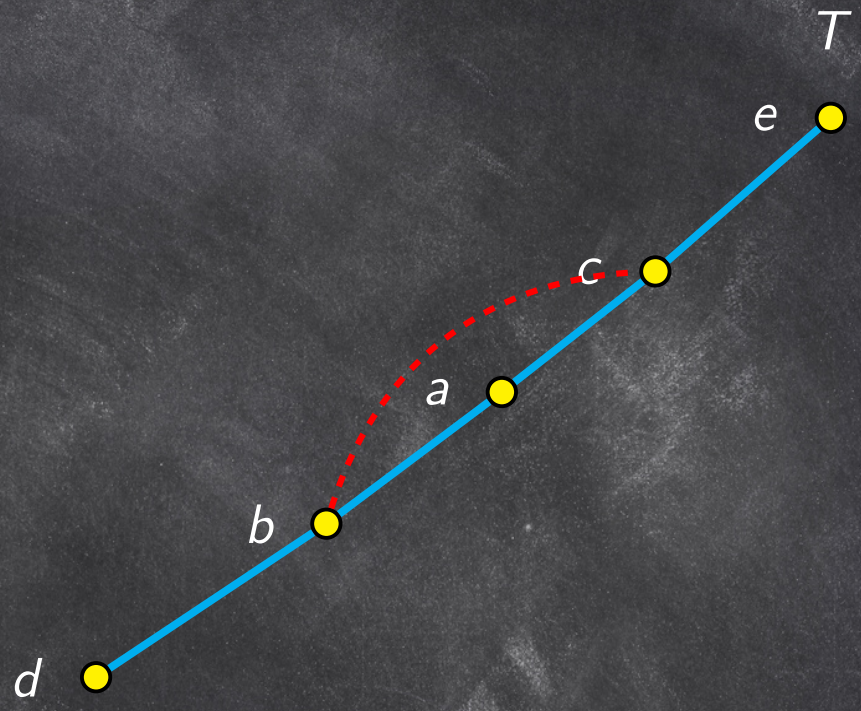
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

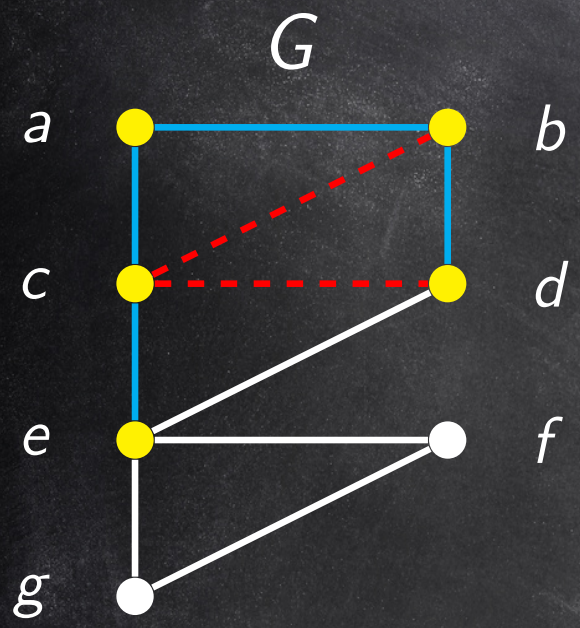
$P(d) \rightarrow N(d) = \{b, c, e\}$

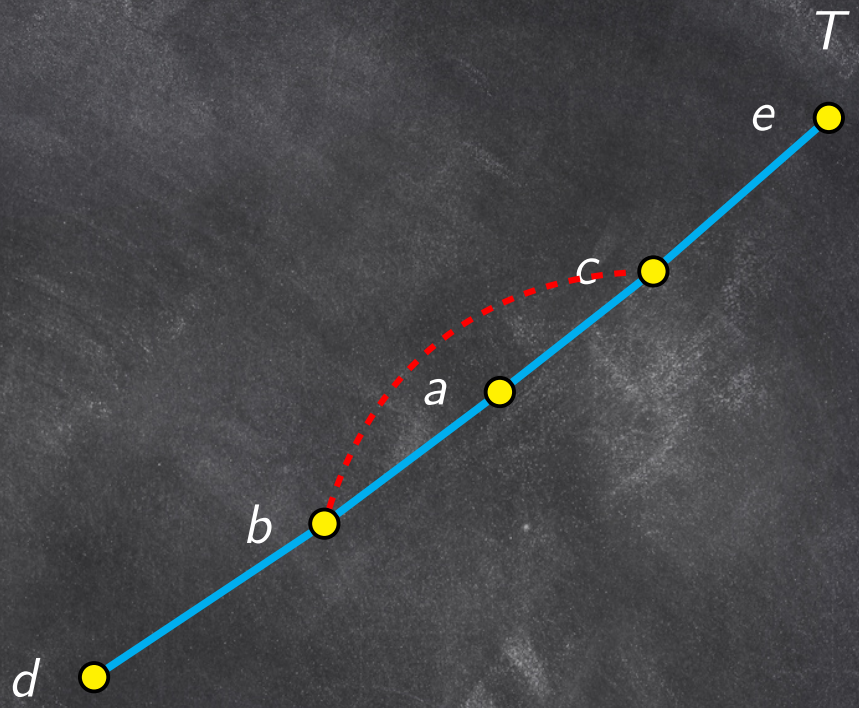
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

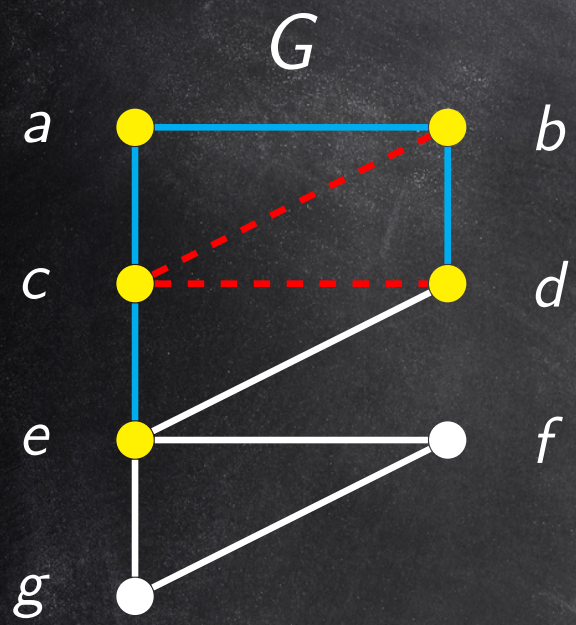
$P(d) \rightarrow N(d) = \{b, c, e\}$

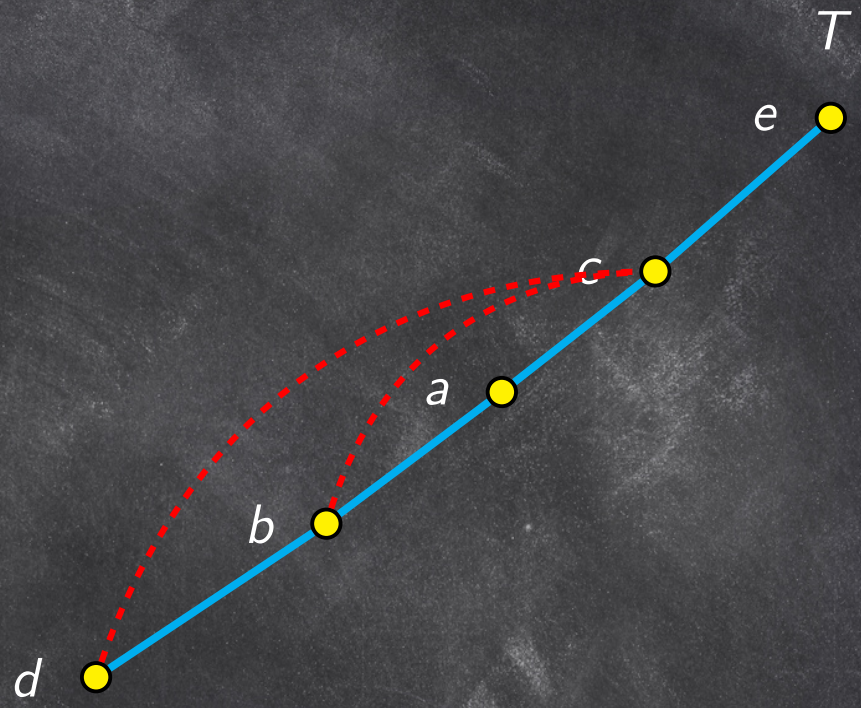
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

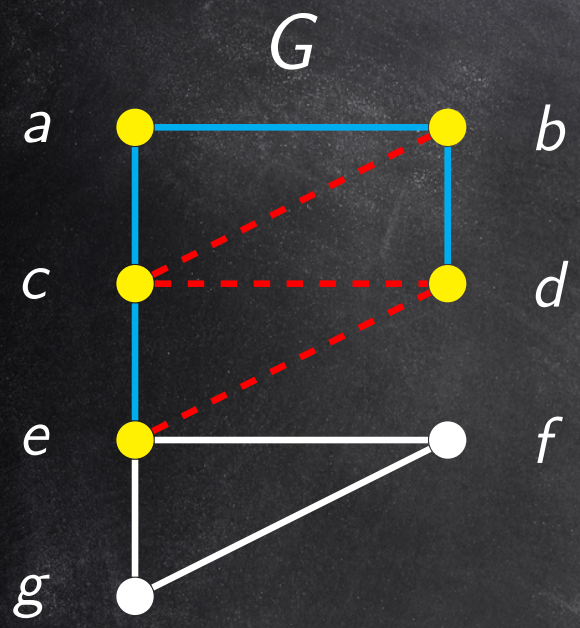
$P(d) \rightarrow N(d) = \{b, c, e\}$

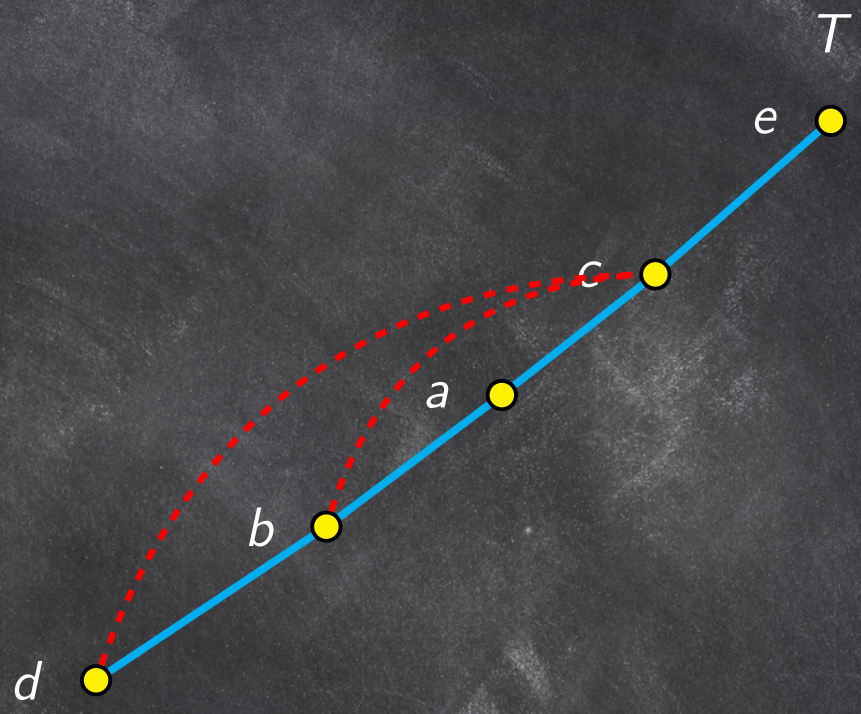
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

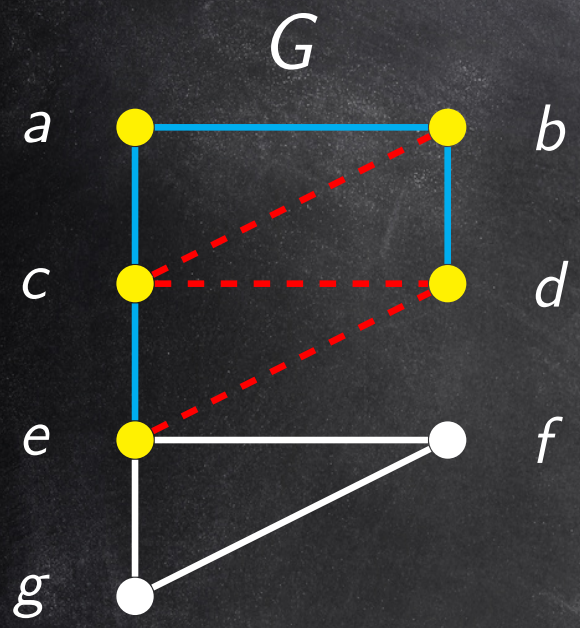
$P(d) \rightarrow N(d) = \{b, c, e\}$

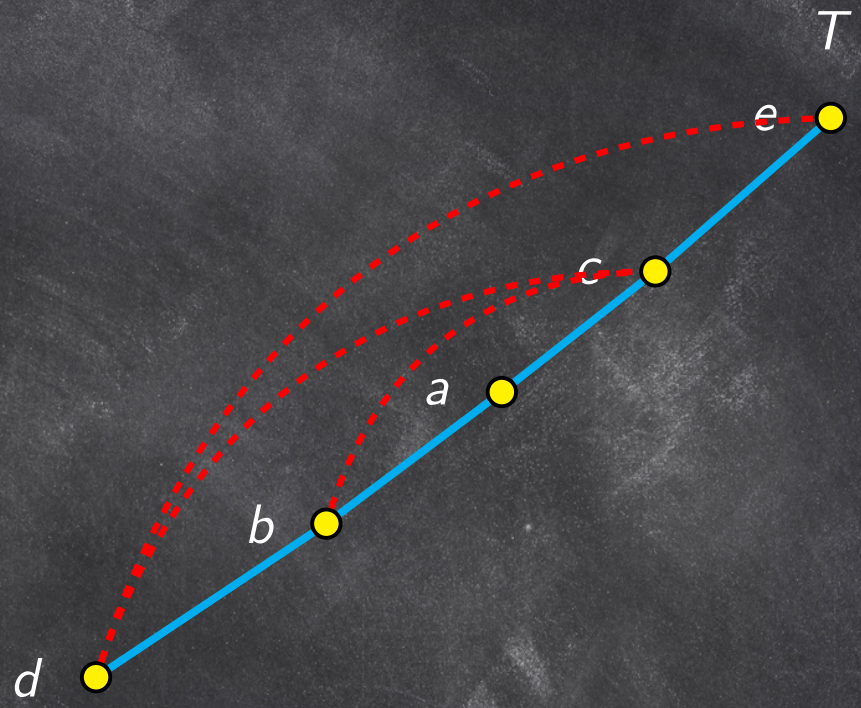
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

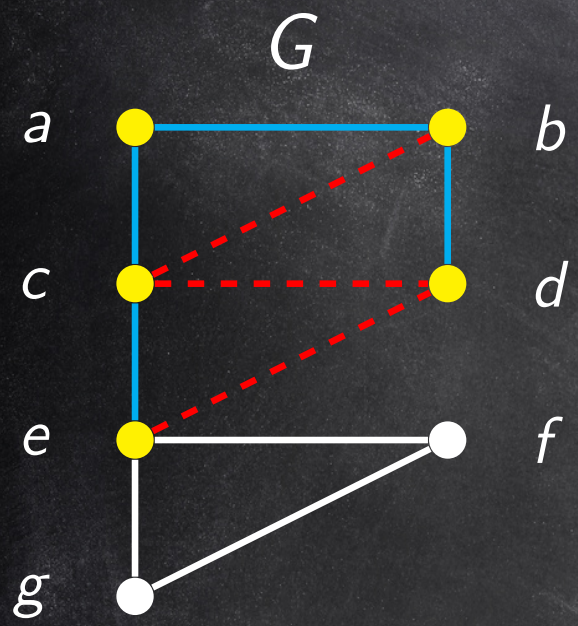
$P(d) \rightarrow N(d) = \{b, c, e\}$

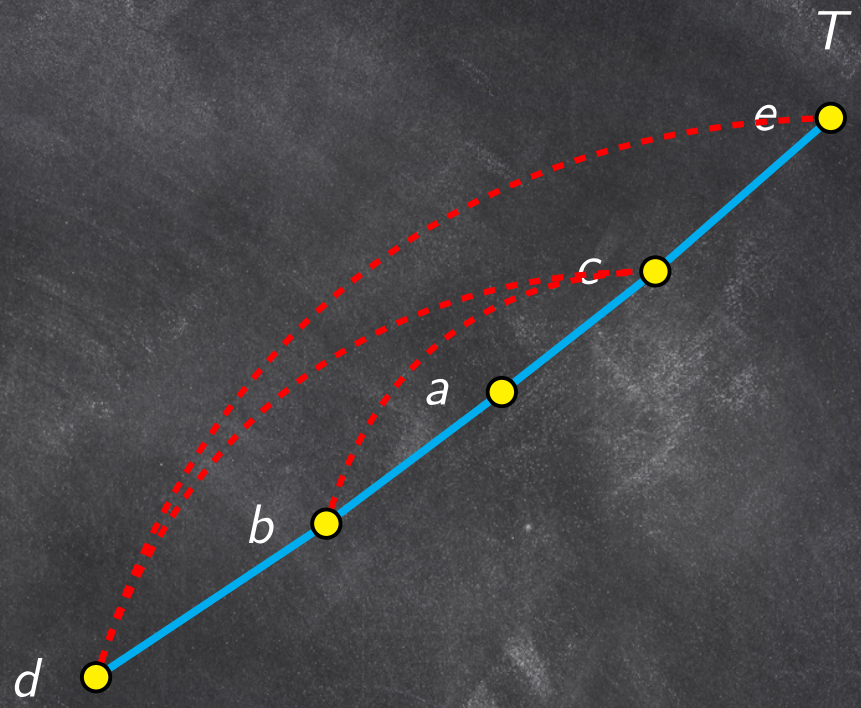
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	0	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

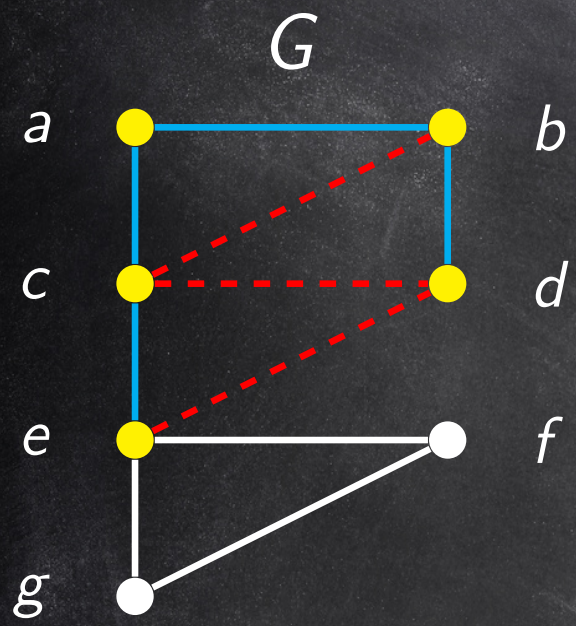
$P(d) \rightarrow N(d) = \{b, c, e\}$

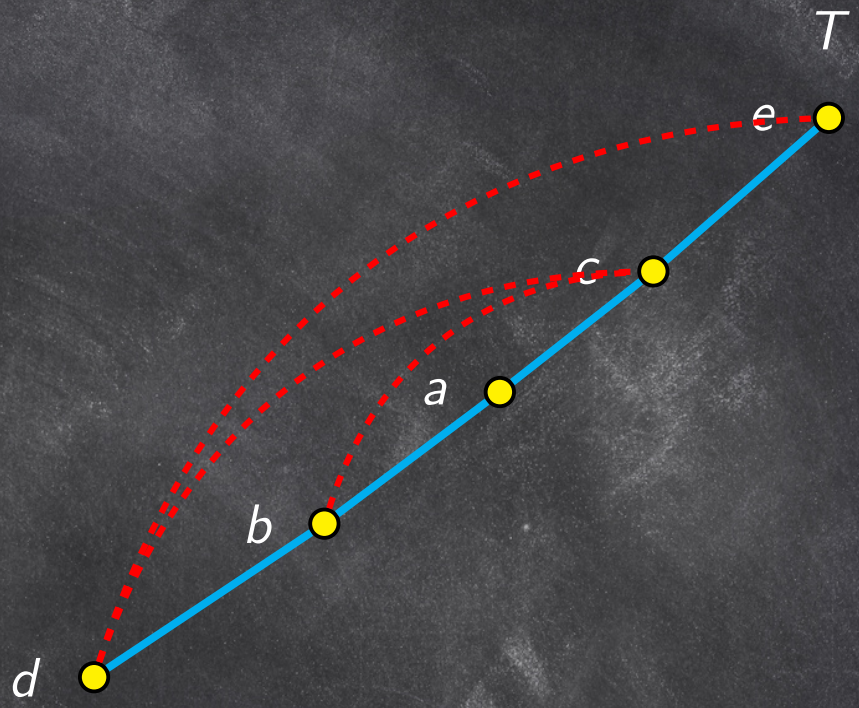
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	0	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

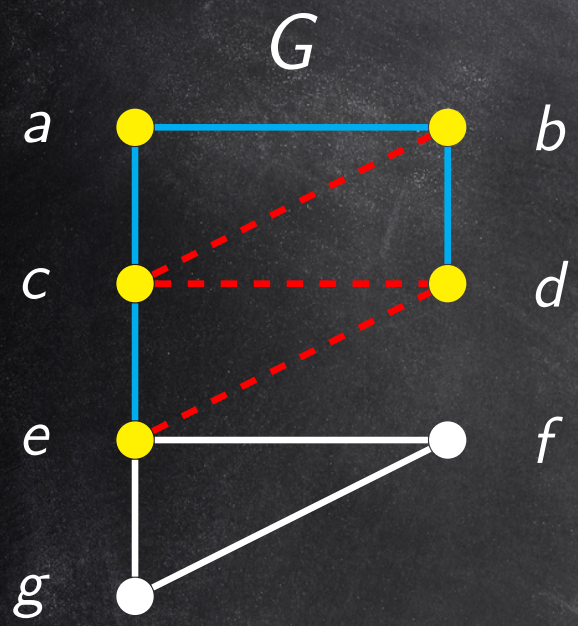
$P(d) \rightarrow N(d) = \{b, c, e\}$

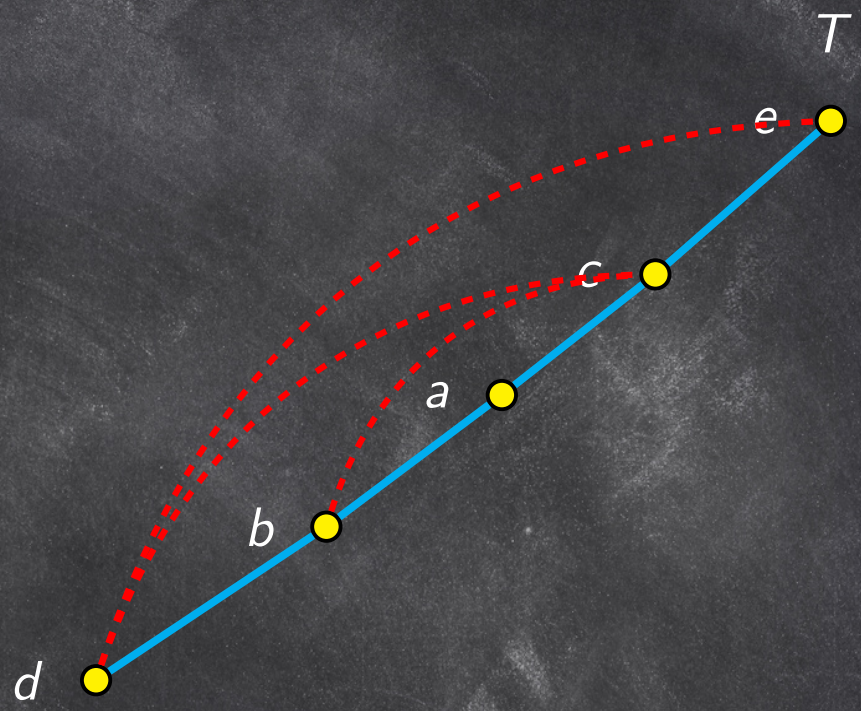
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

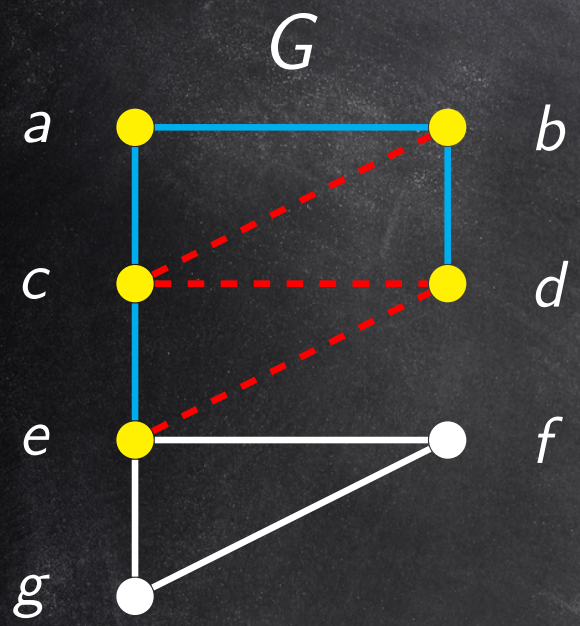
$P(d) \rightarrow N(d) = \{b, c, e\}$

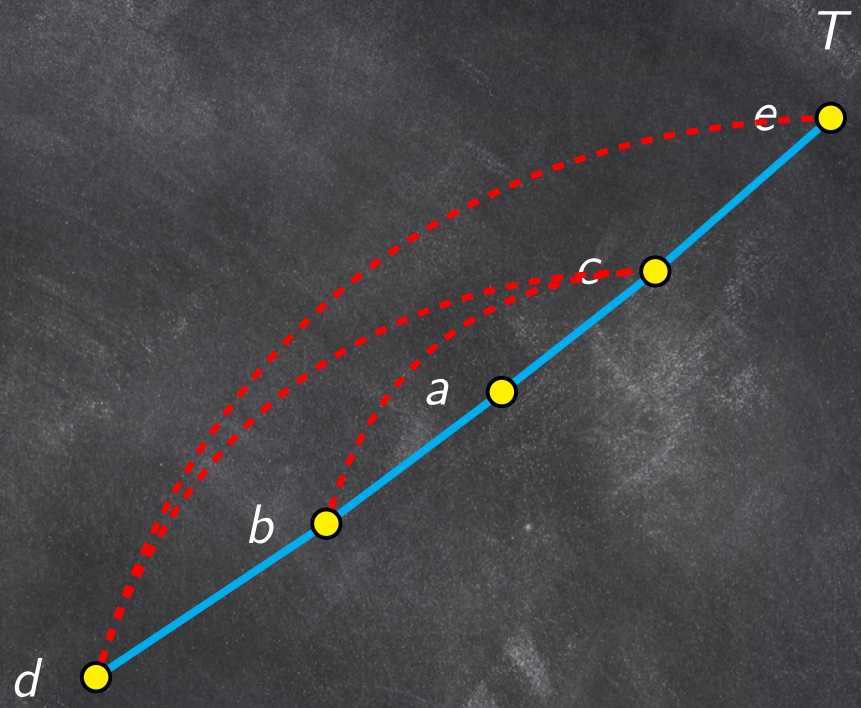
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	0	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

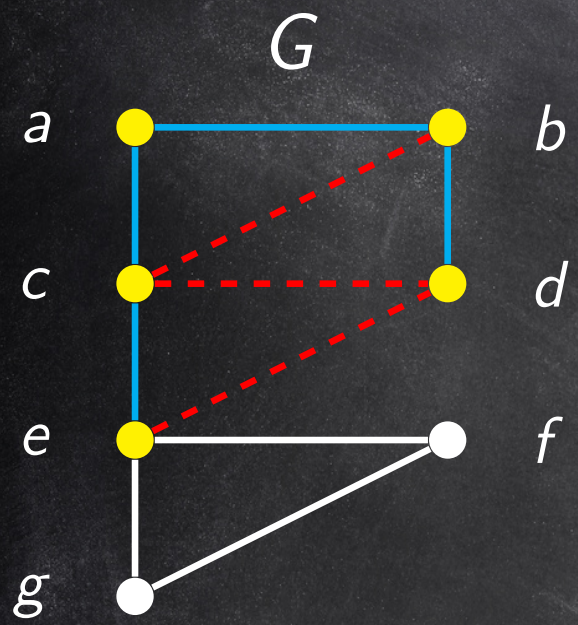
$P(d) \rightarrow N(d) = \{b, c, e\}$

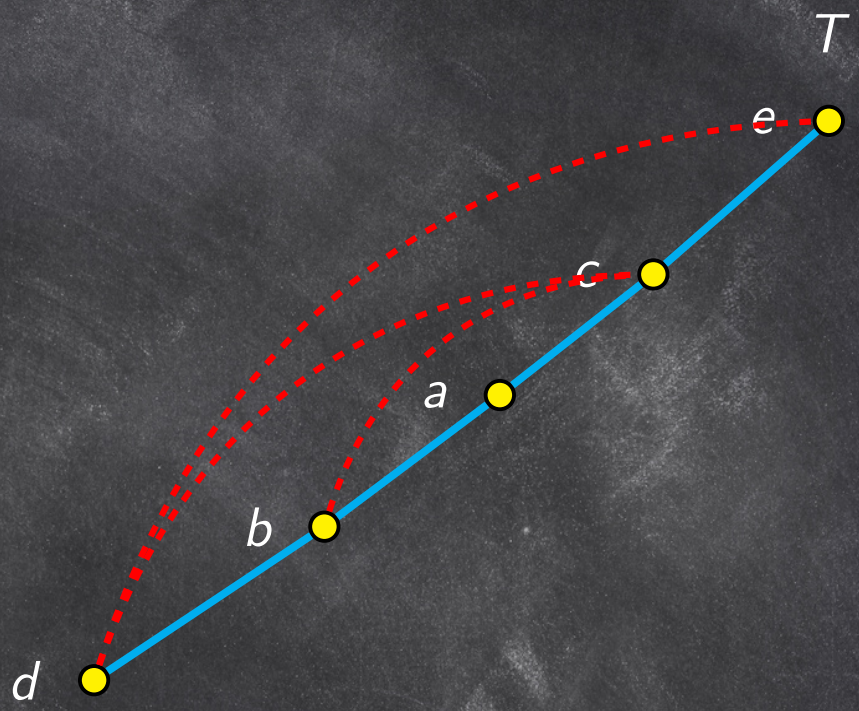
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

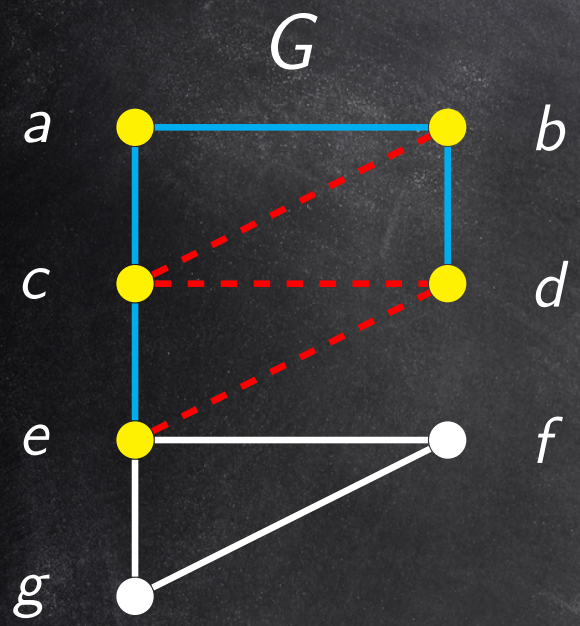
$P(d) \rightarrow N(d) = \{b, c, e\}$

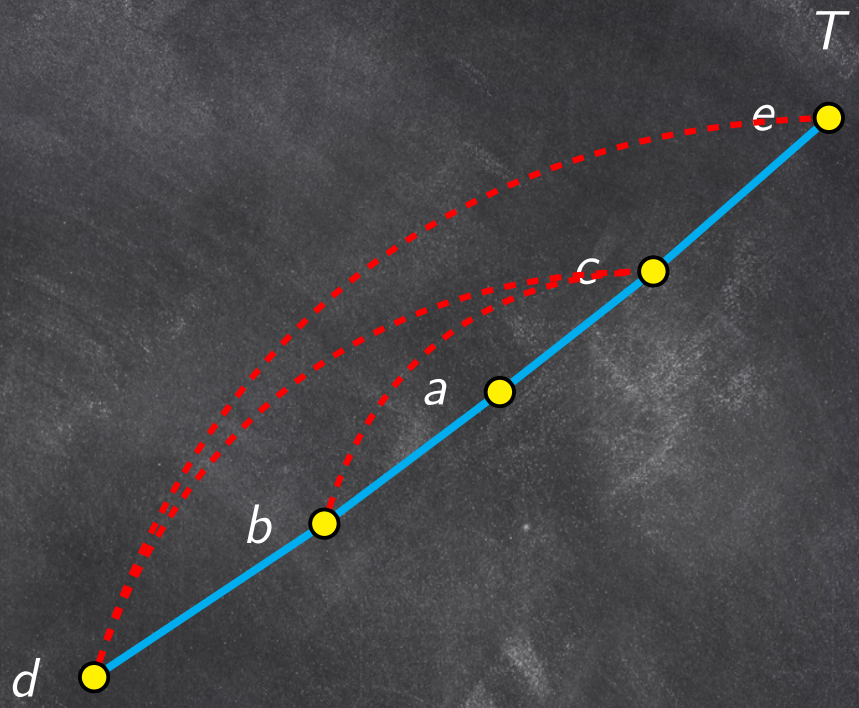
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

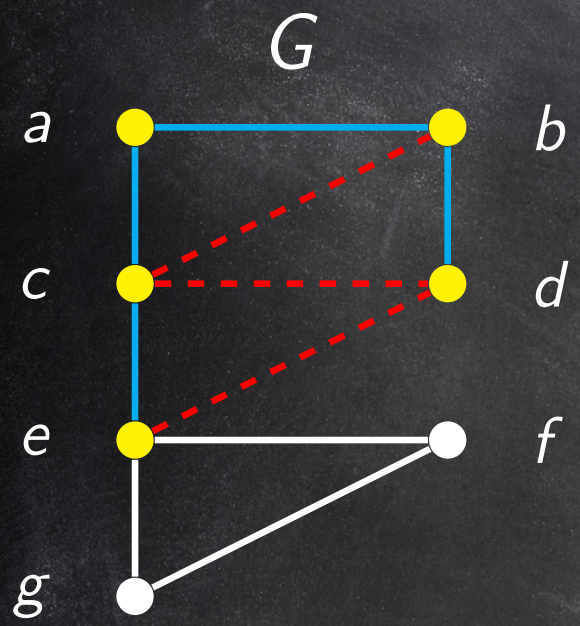
$P(d) \rightarrow N(d) = \{b, c, e\}$

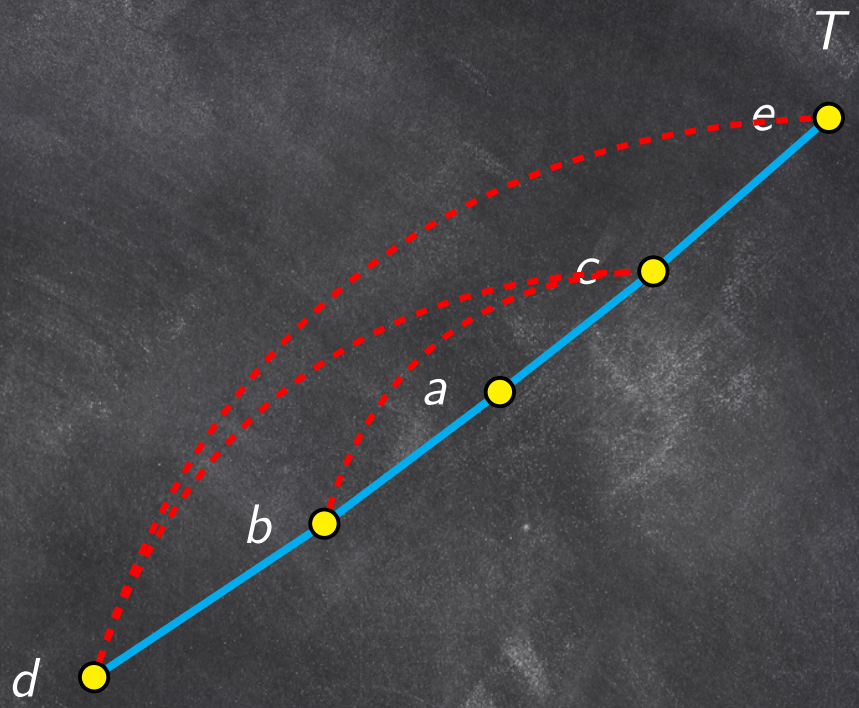
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

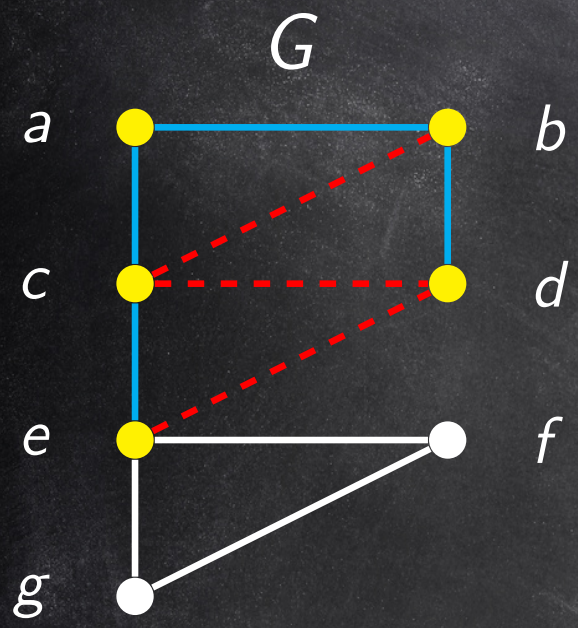
$P(d) \rightarrow N(d) = \{b, c, e\}$

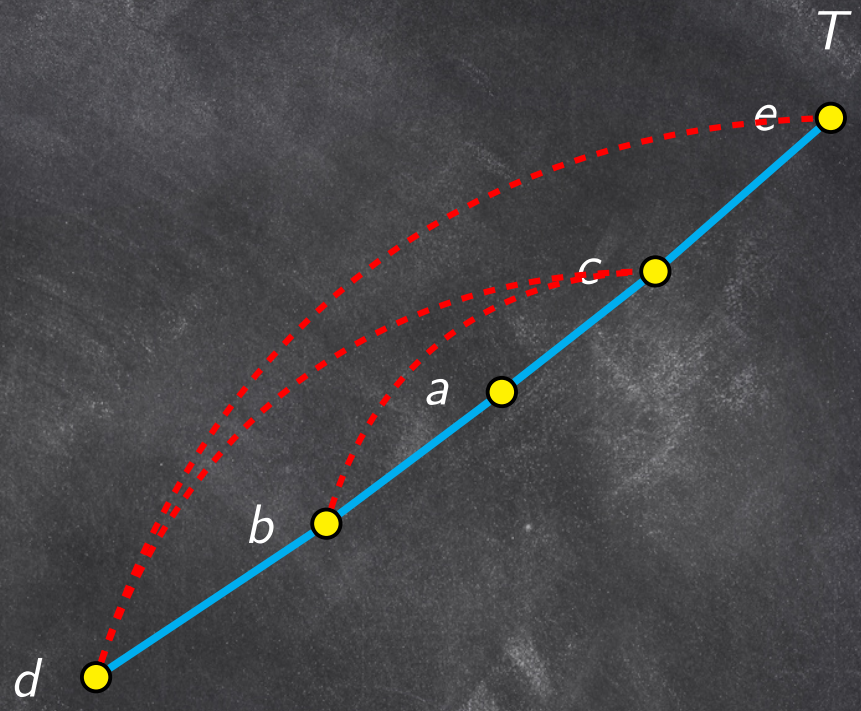
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	0	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

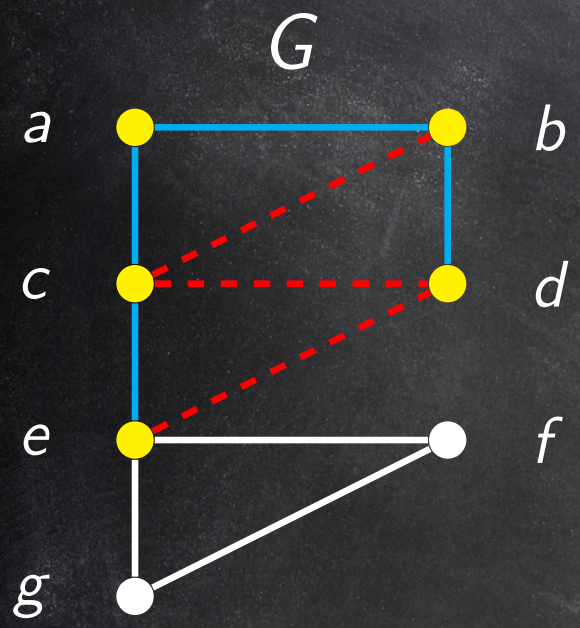
$P(d) \rightarrow N(d) = \{b, c, e\}$

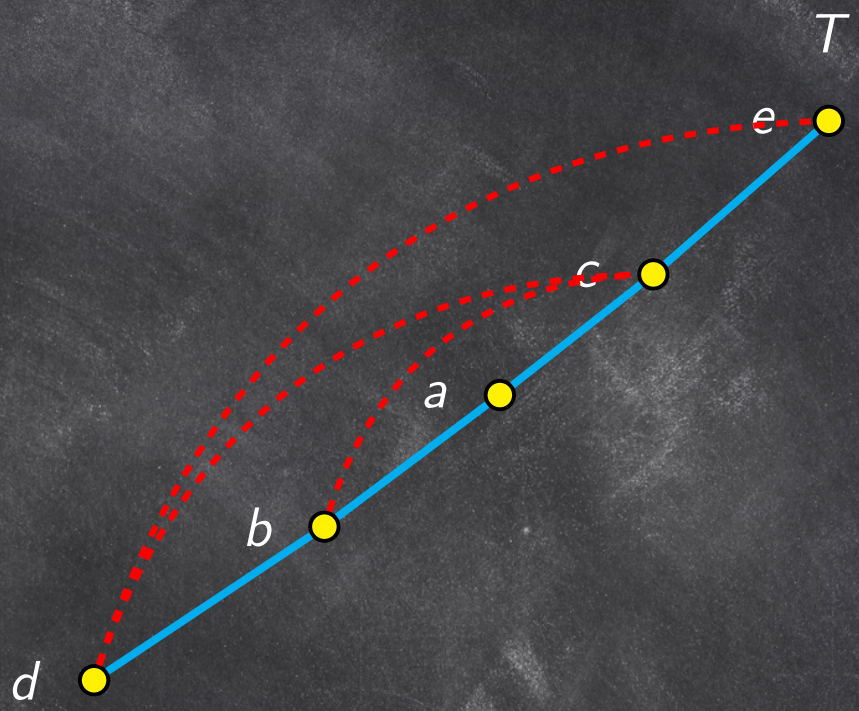
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

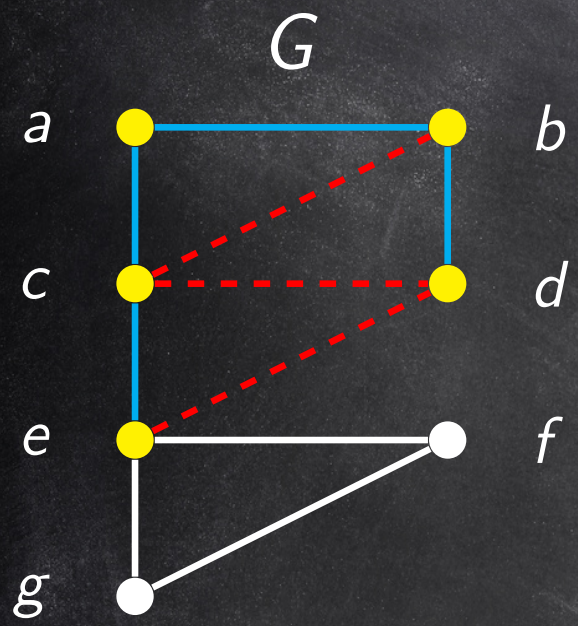
$P(d) \rightarrow N(d) = \{b, c, e\}$

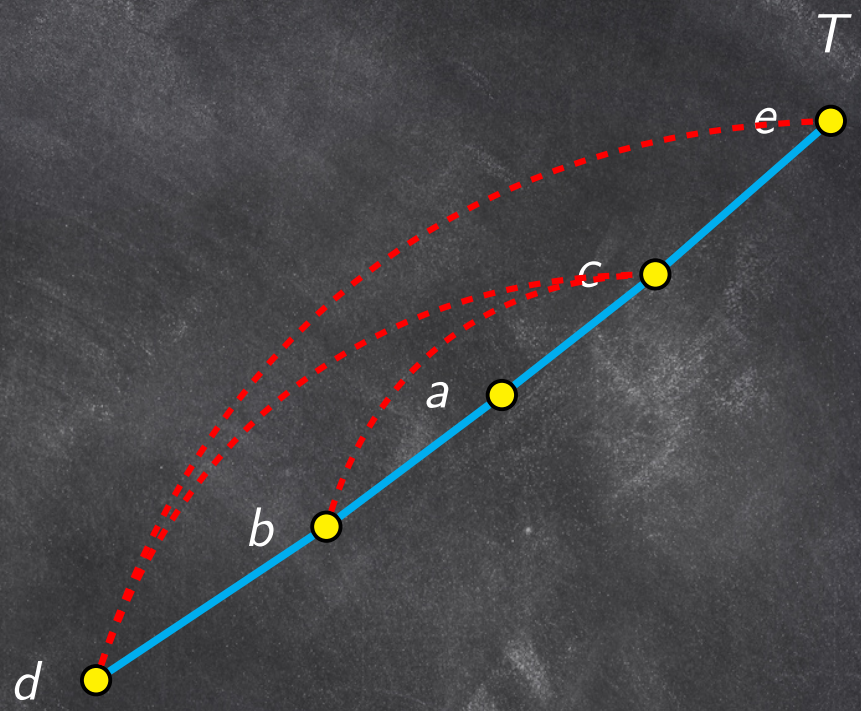
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

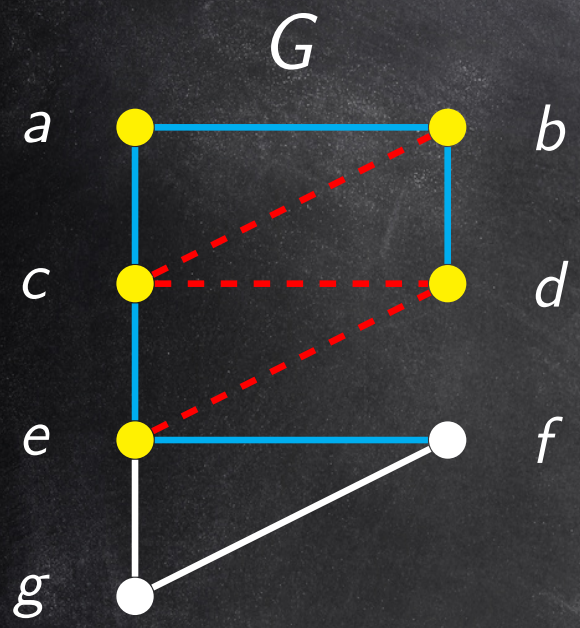
$P(d) \rightarrow N(d) = \{b, c, e\}$

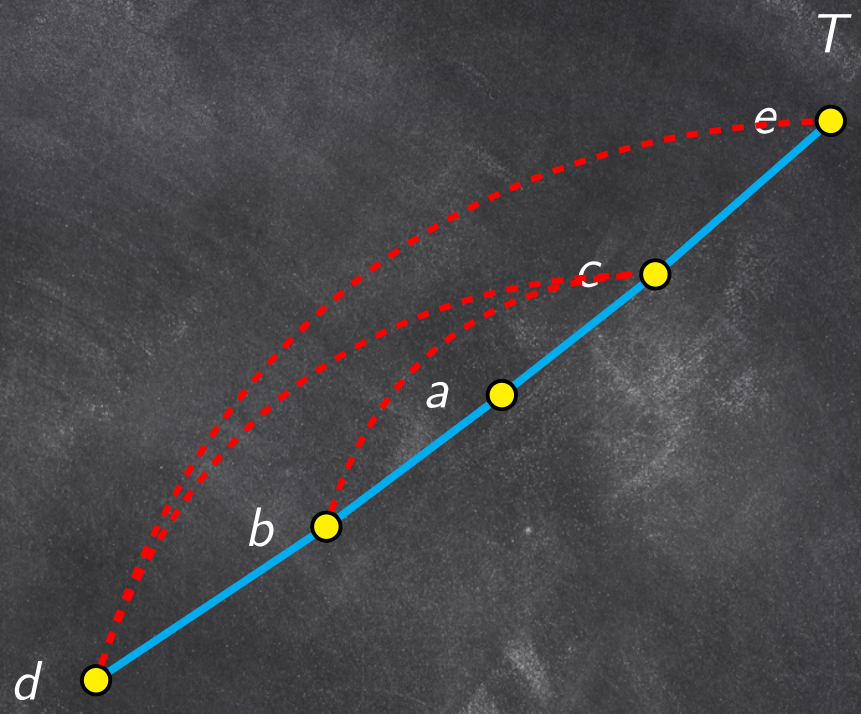
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

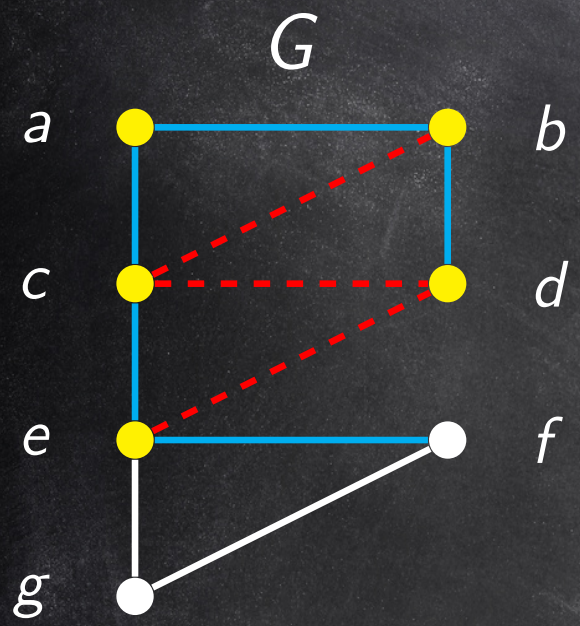
$P(d) \rightarrow N(d) = \{b, c, e\}$

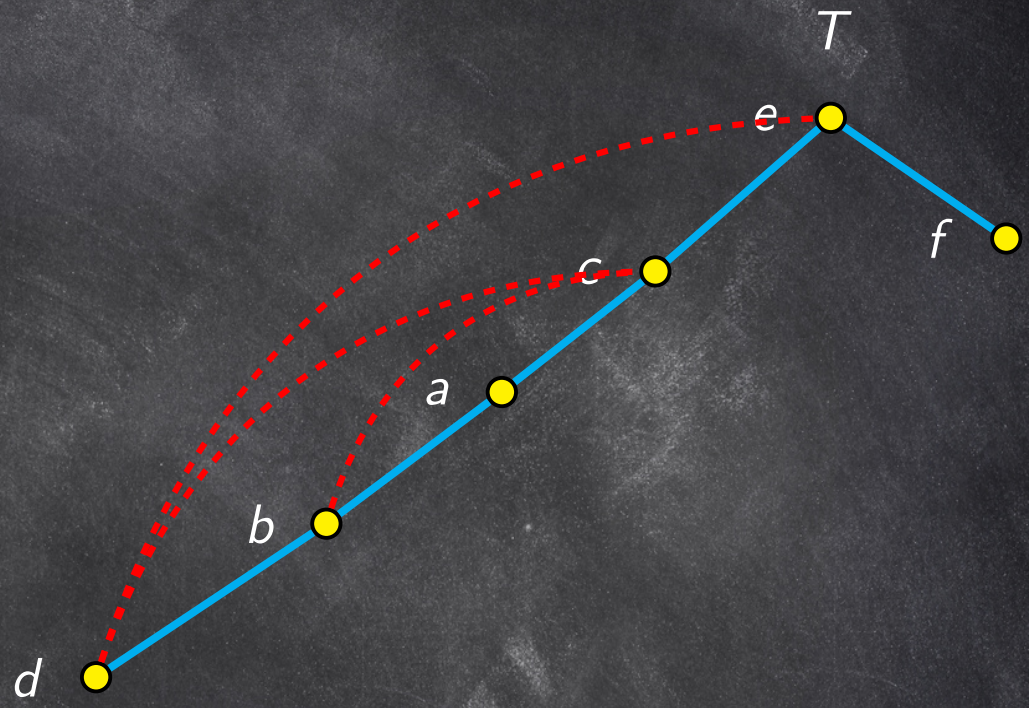
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

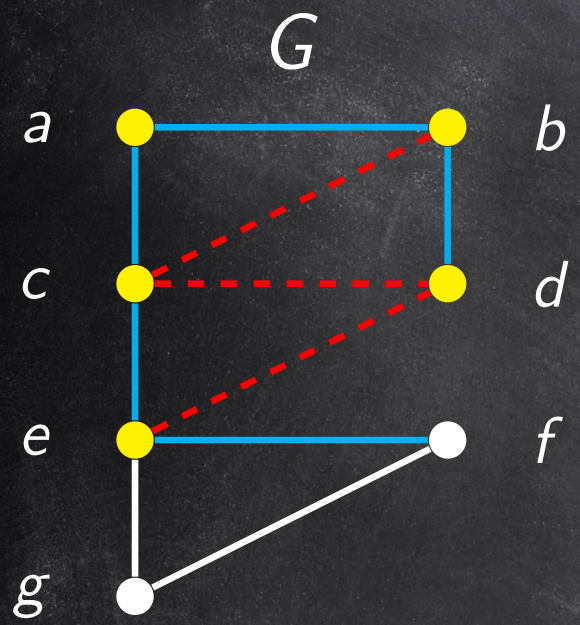
$P(d) \rightarrow N(d) = \{b, c, e\}$

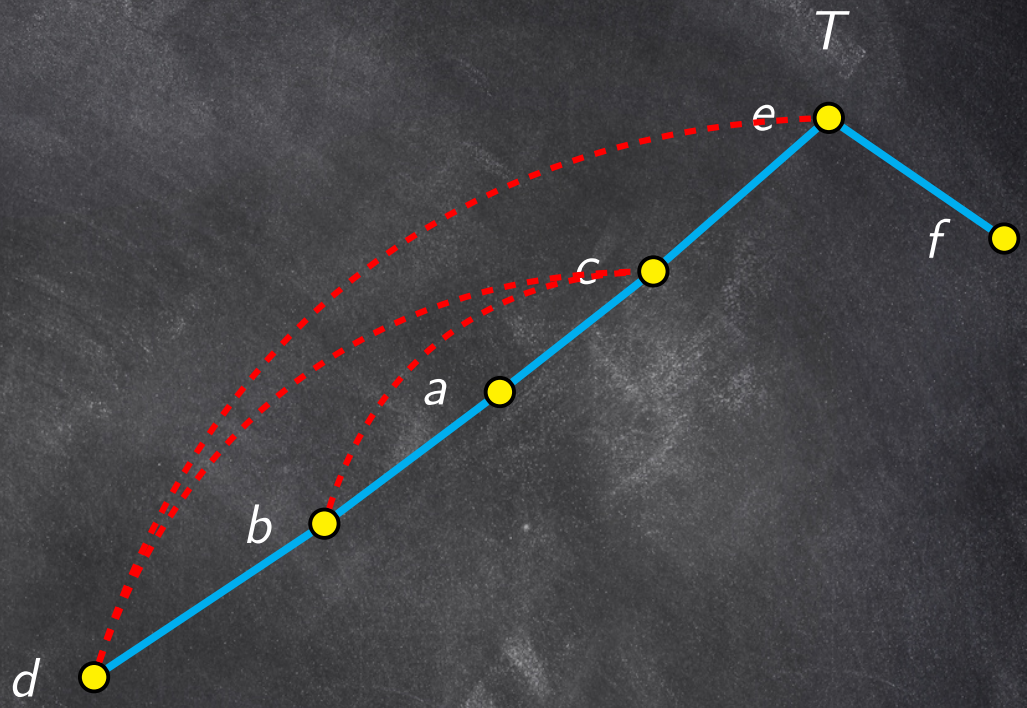
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

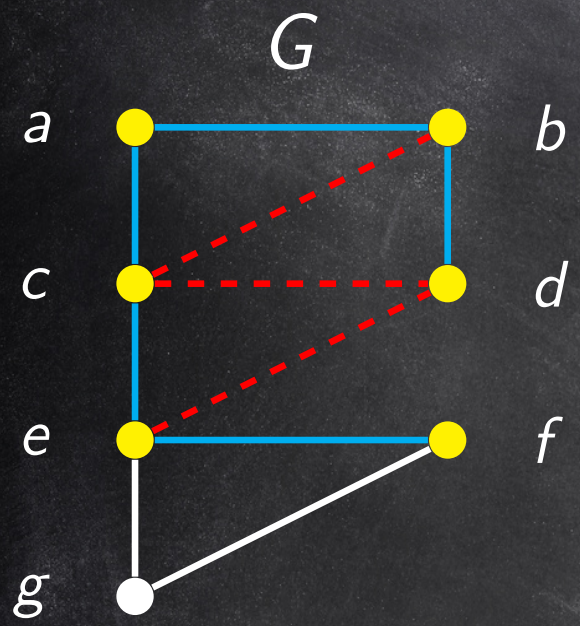
$P(f)$

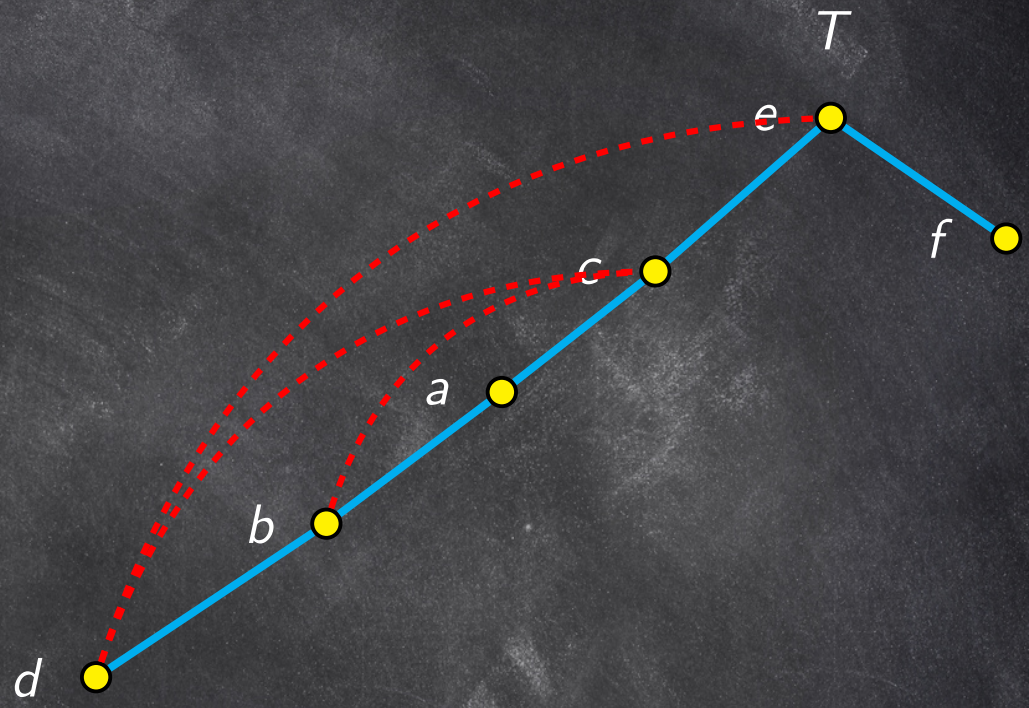
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

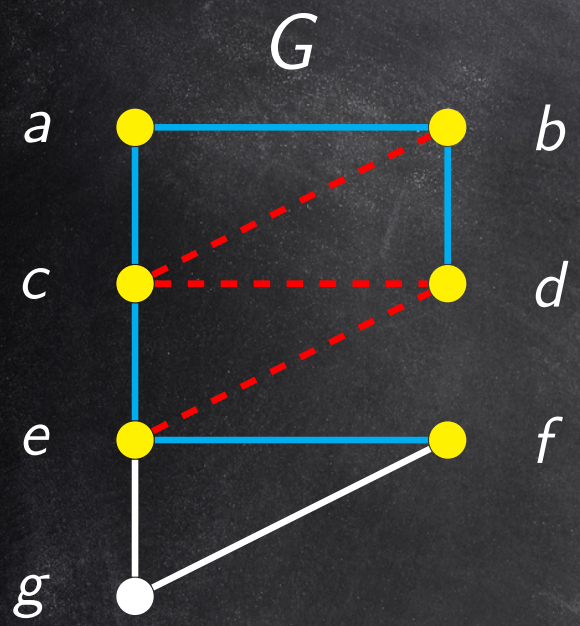
$P(f)$

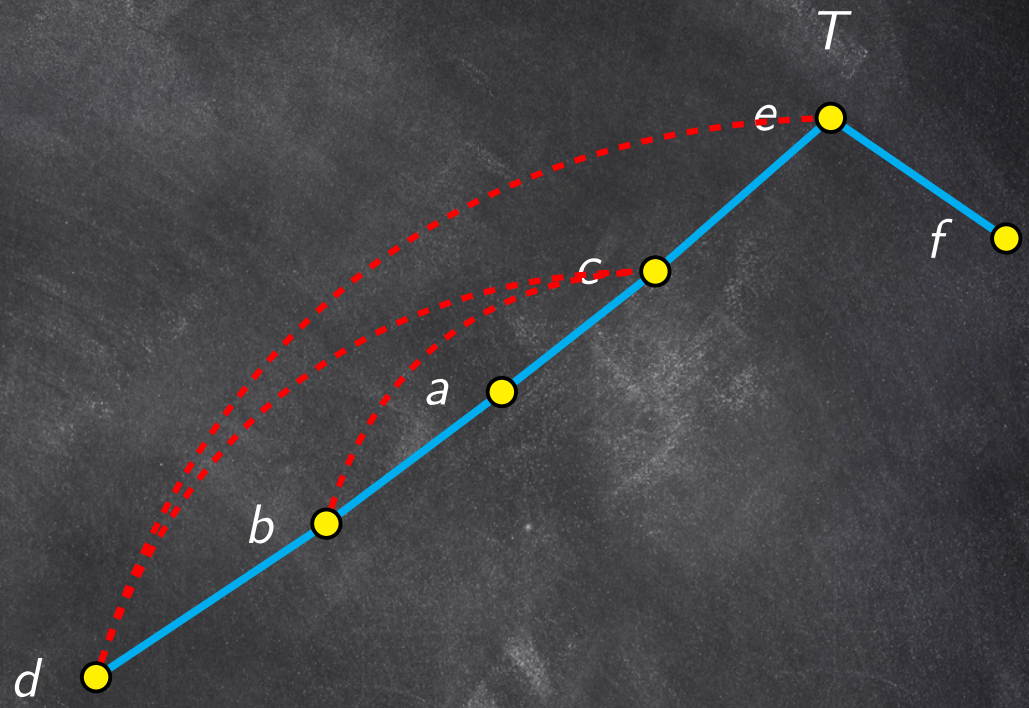
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	0	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

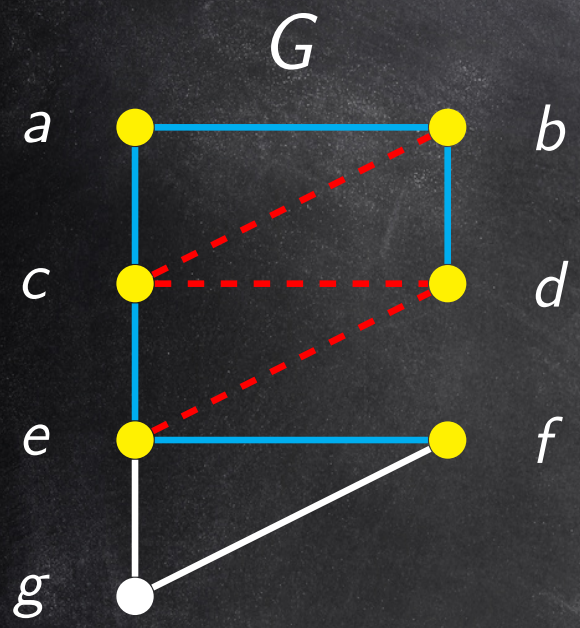
$P(f)$

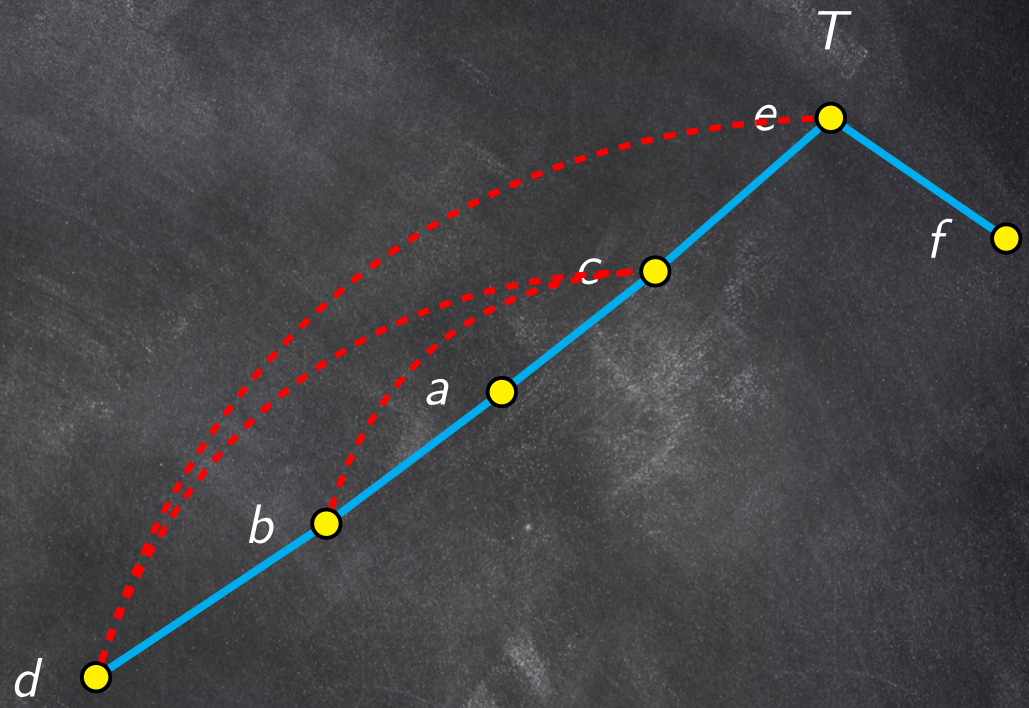
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

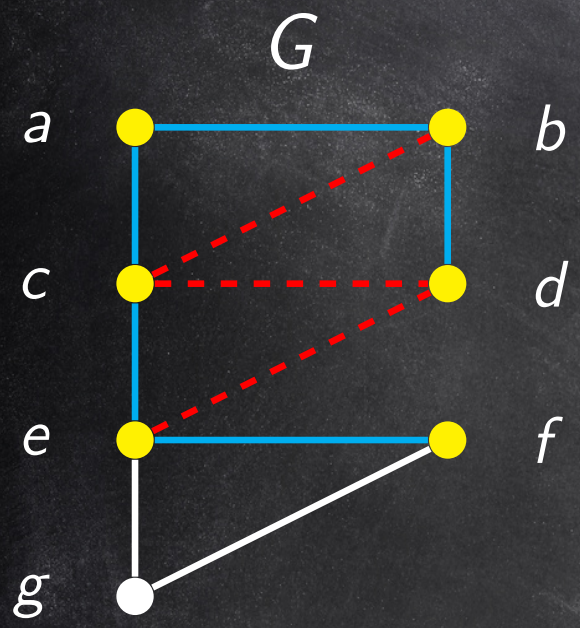
$P(f) \rightarrow N(f) = \{e, g\}$

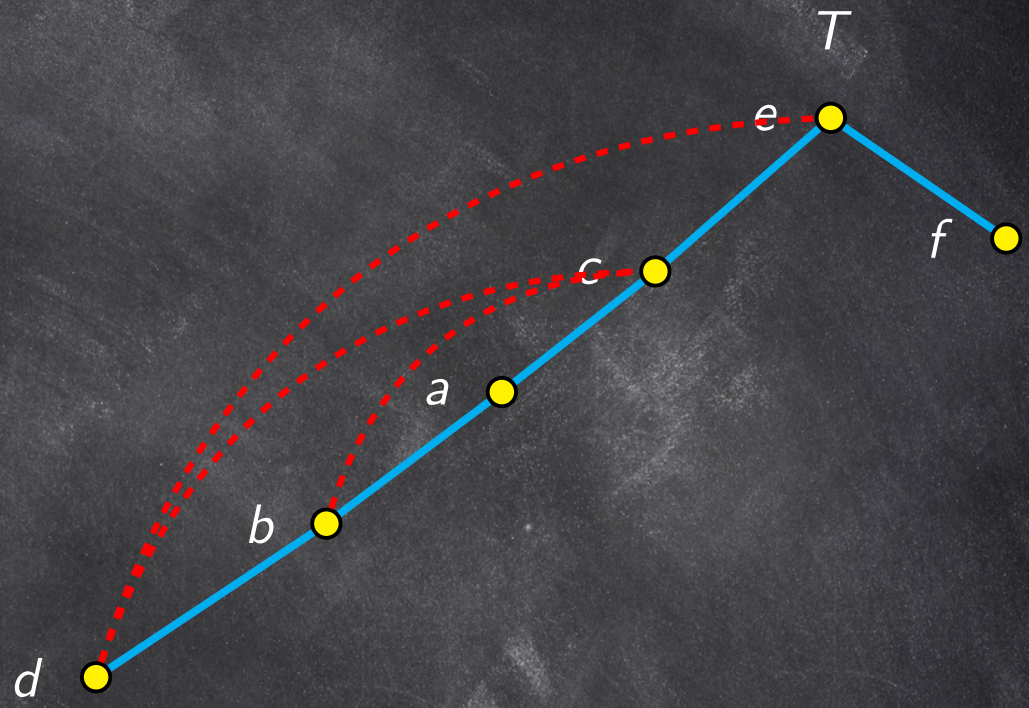
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

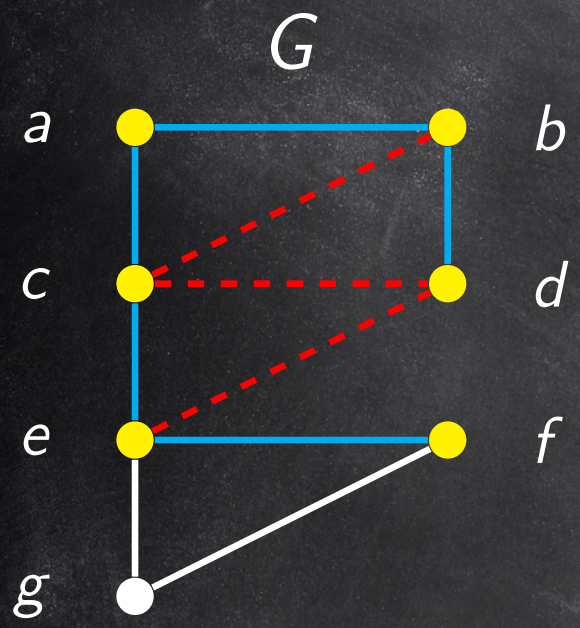
$P(f) \rightarrow N(f) = \{e, g\}$

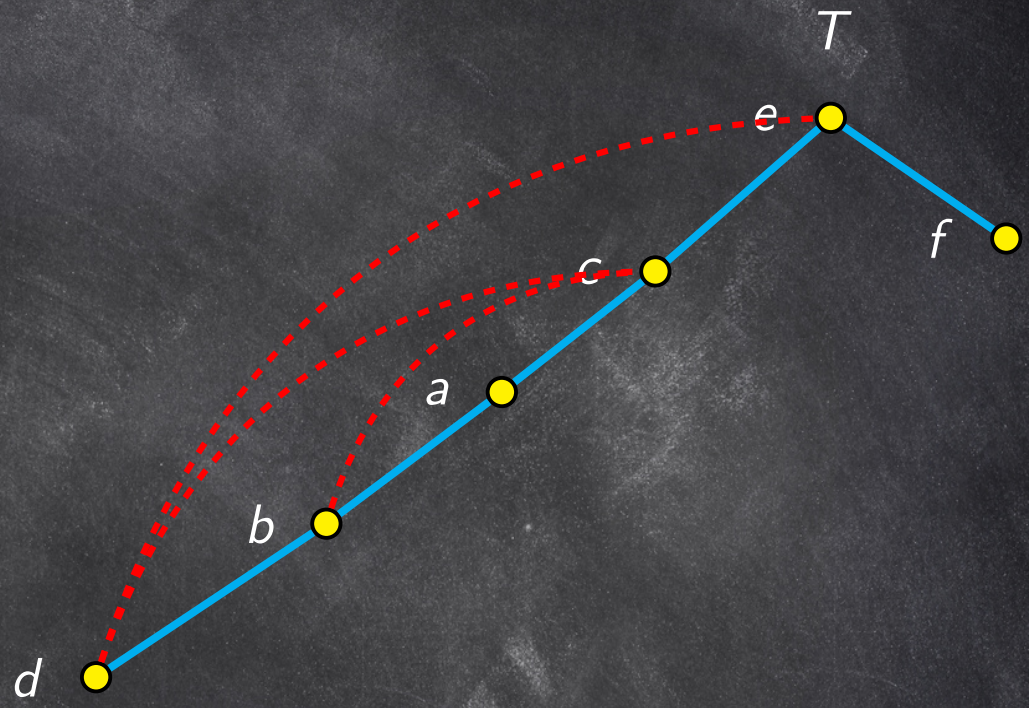
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

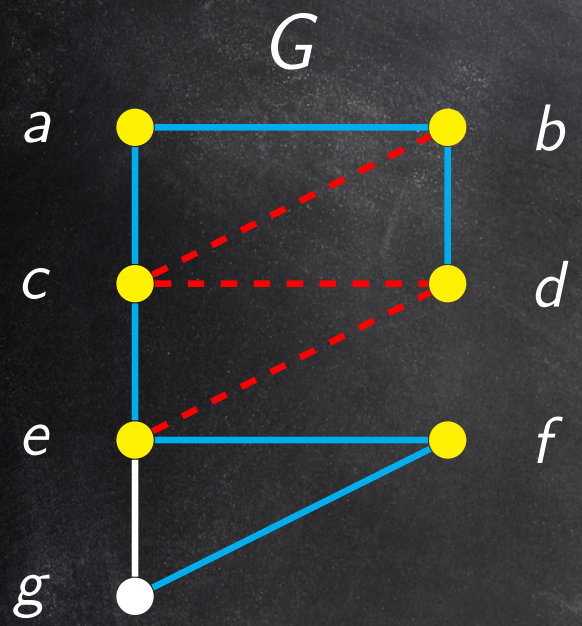
$P(f) \rightarrow N(f) = \{e, g\}$

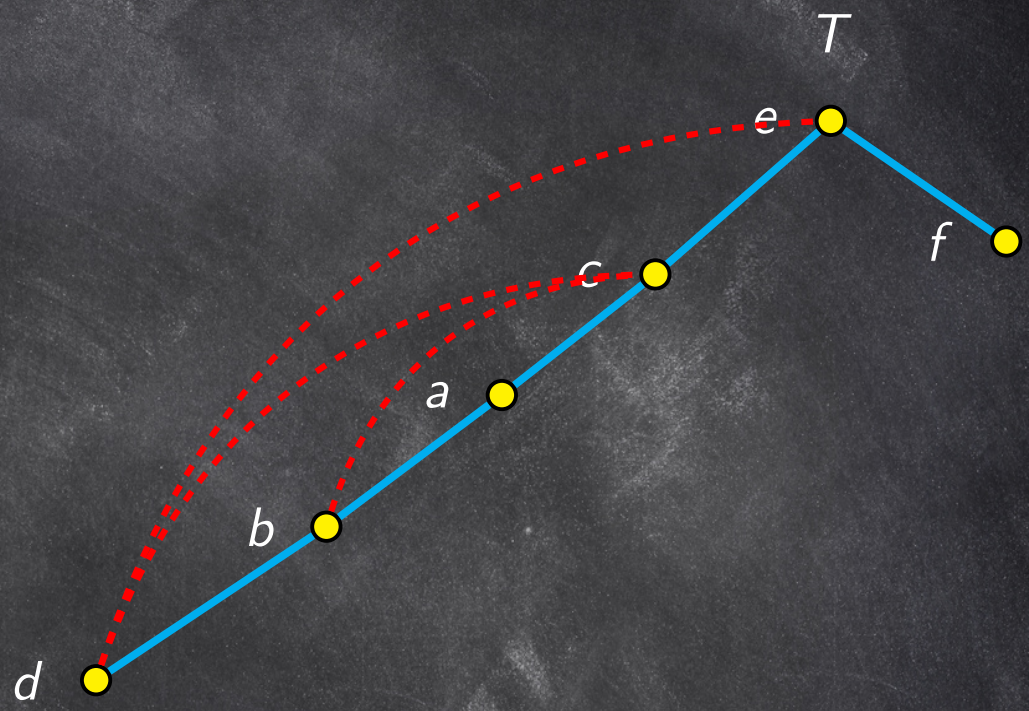
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

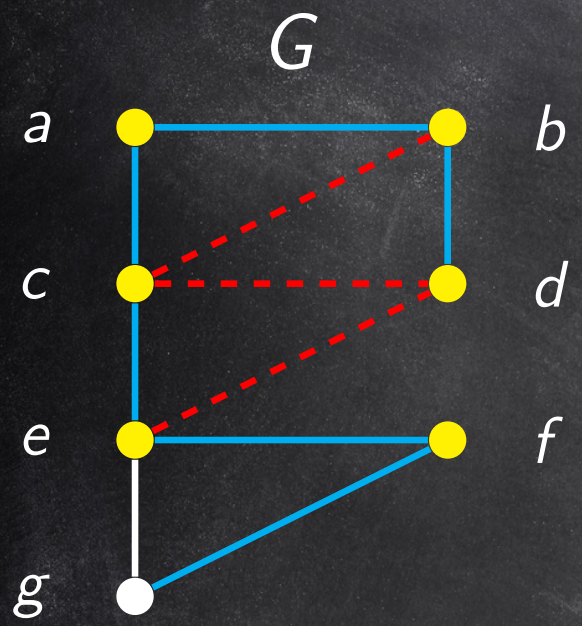
$P(f) \rightarrow N(f) = \{e, g\}$

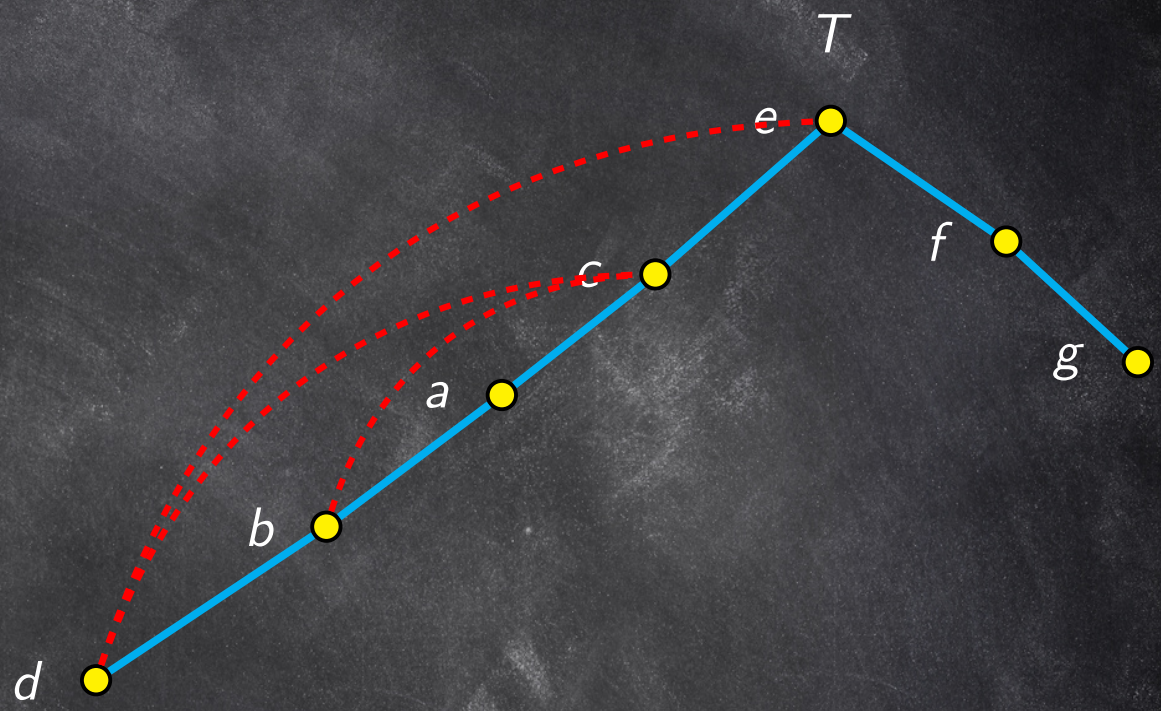
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

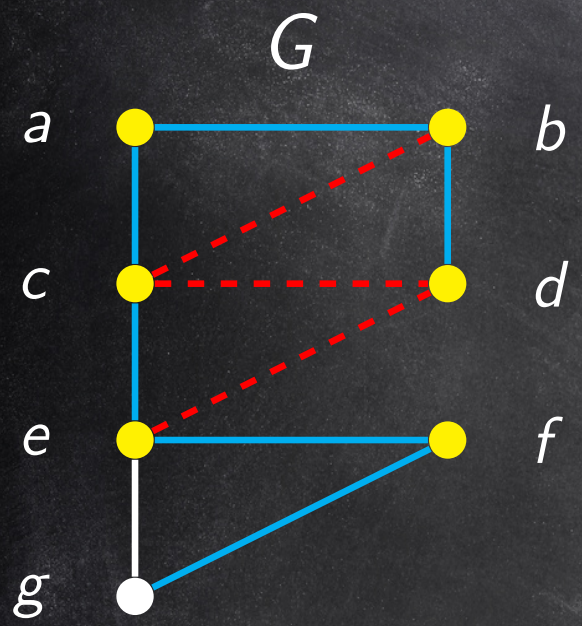
$P(f) \rightarrow N(f) = \{e, g\}$

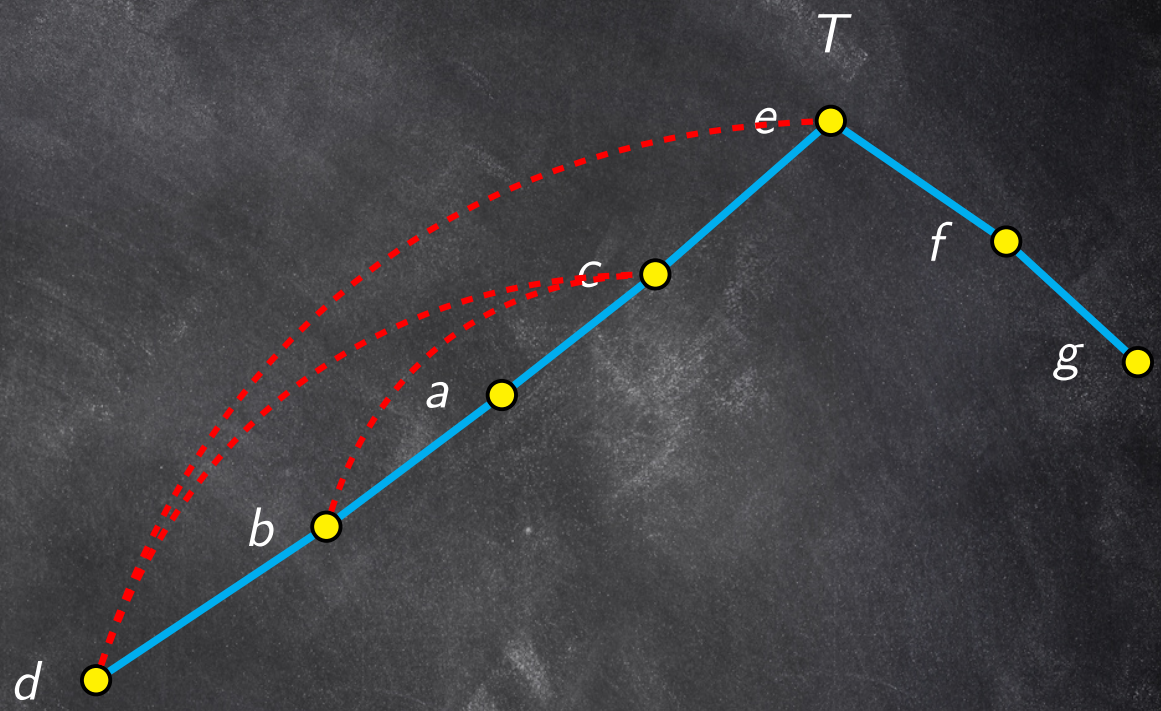
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

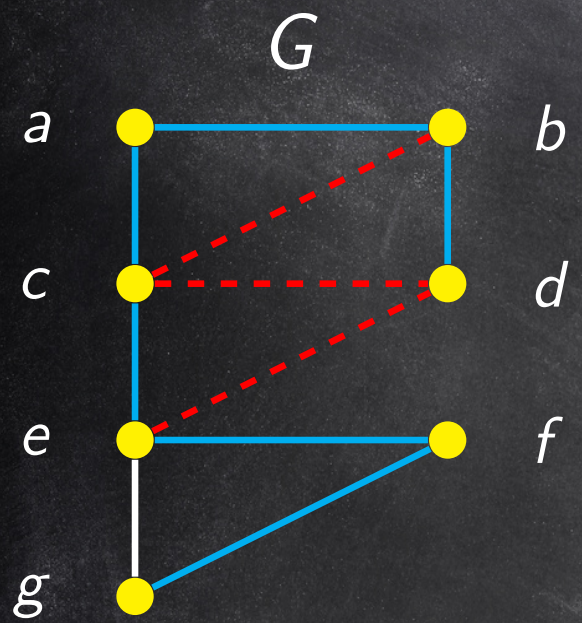
$P(g)$

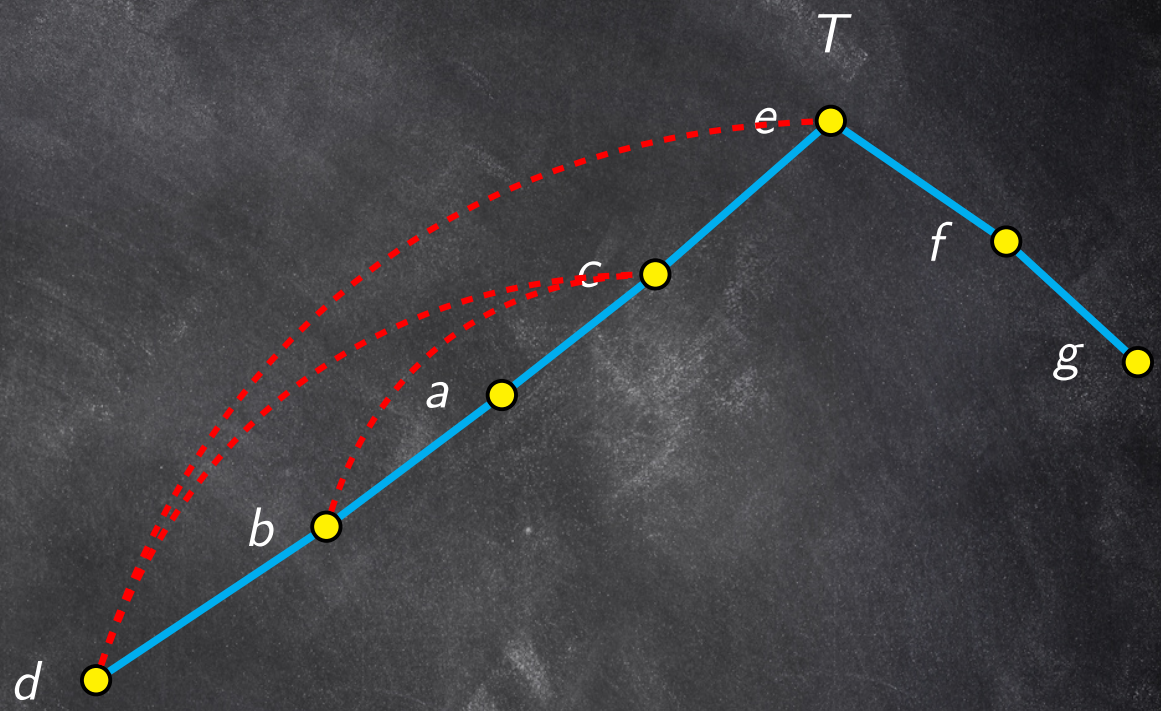
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

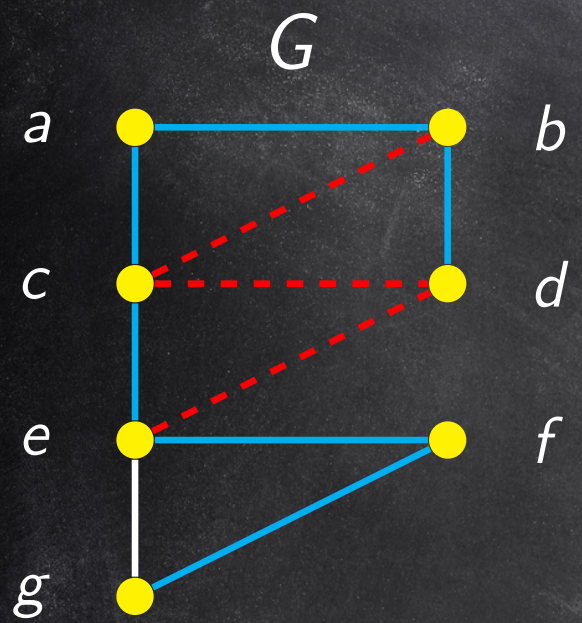
$P(g)$

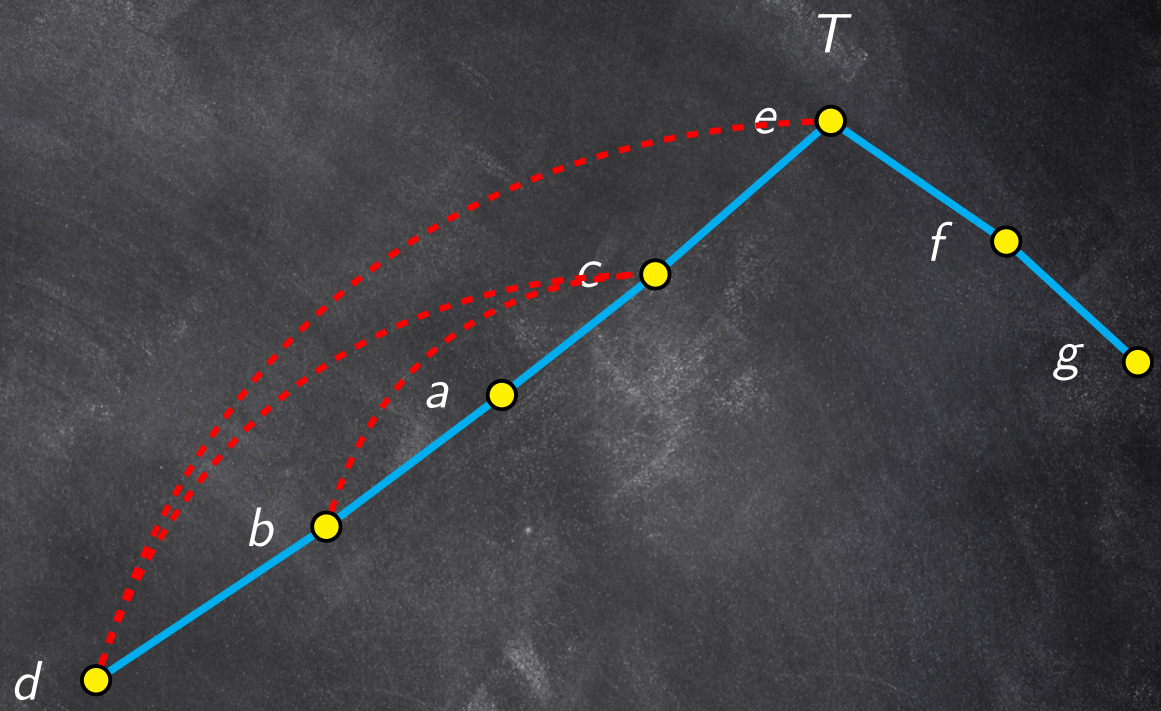
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	0
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

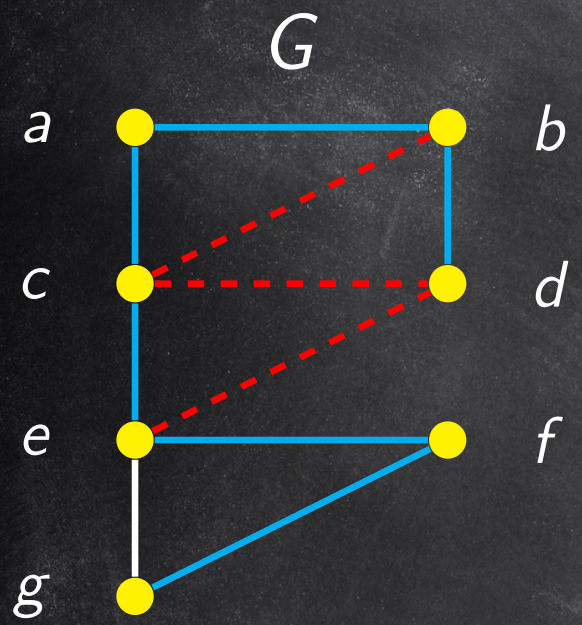
$P(g)$

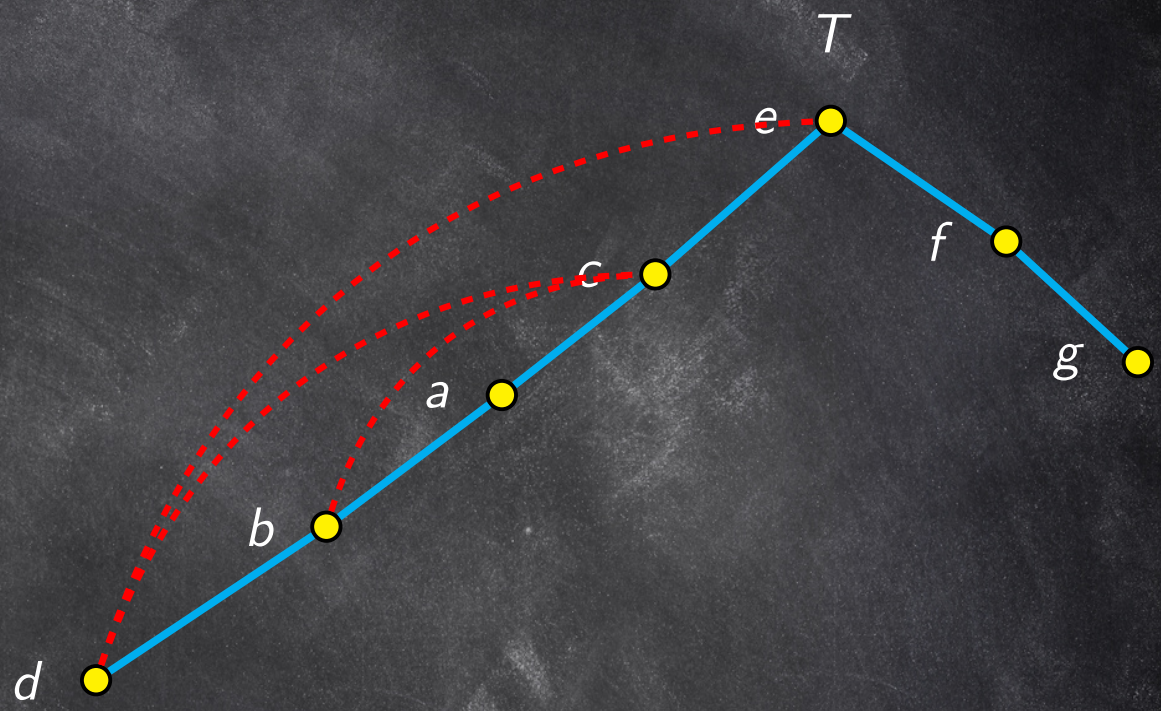
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

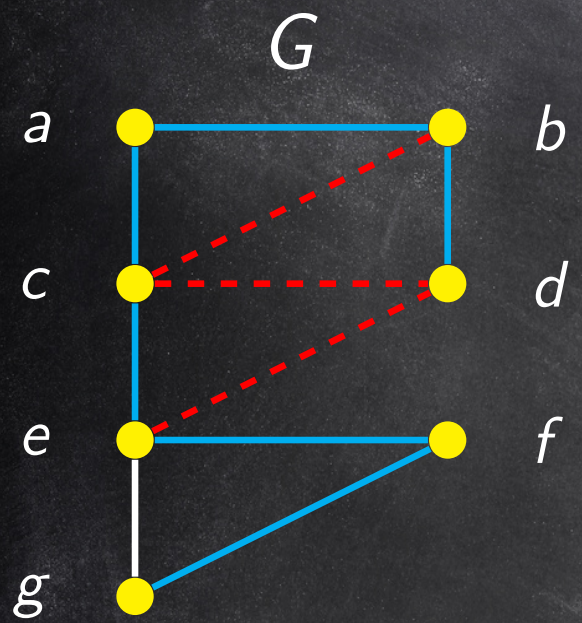
$P(g) \rightarrow N(g) = \{e, f\}$

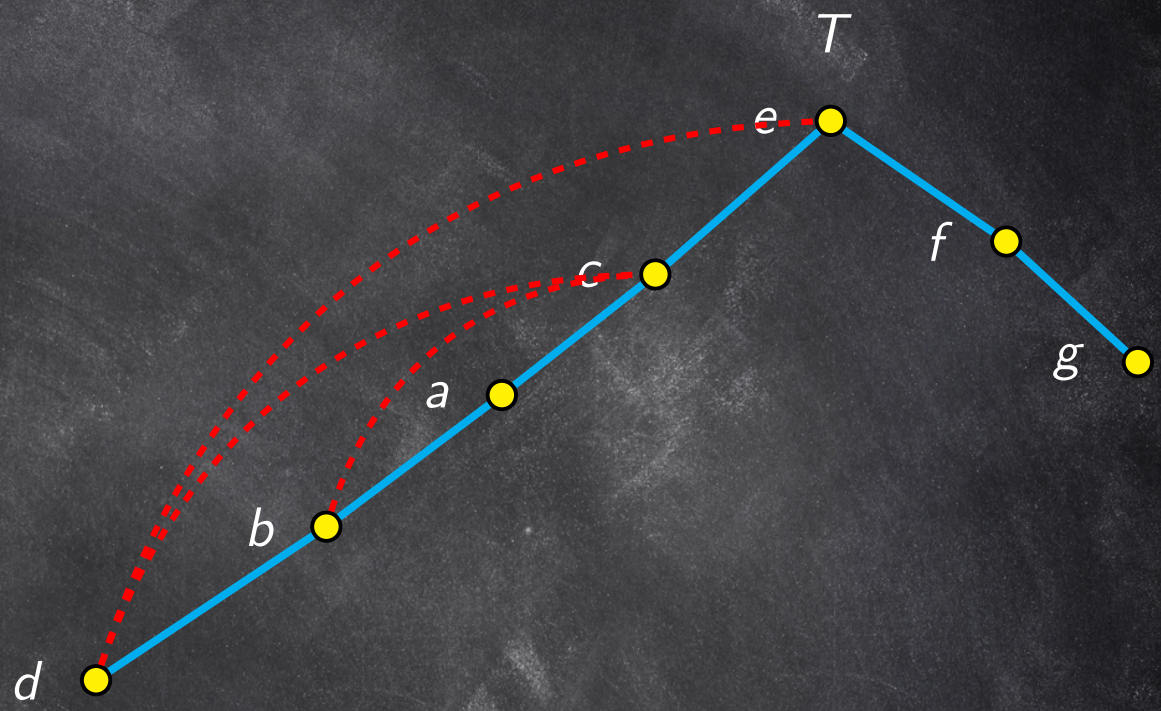
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

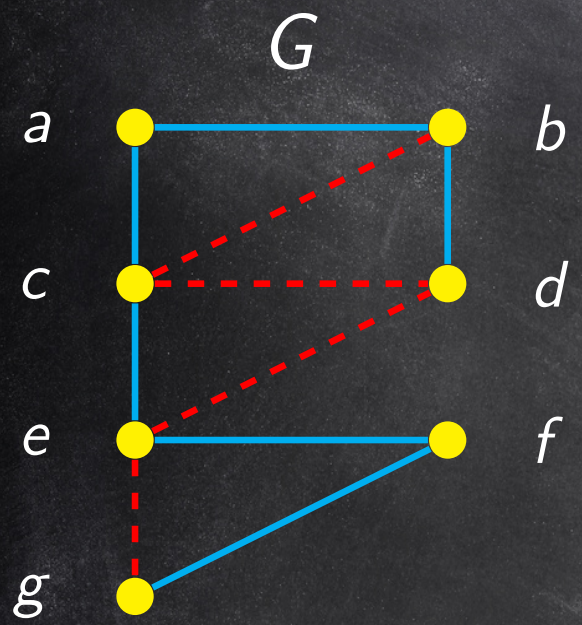
$P(g) \rightarrow N(g) = \{e, f\}$

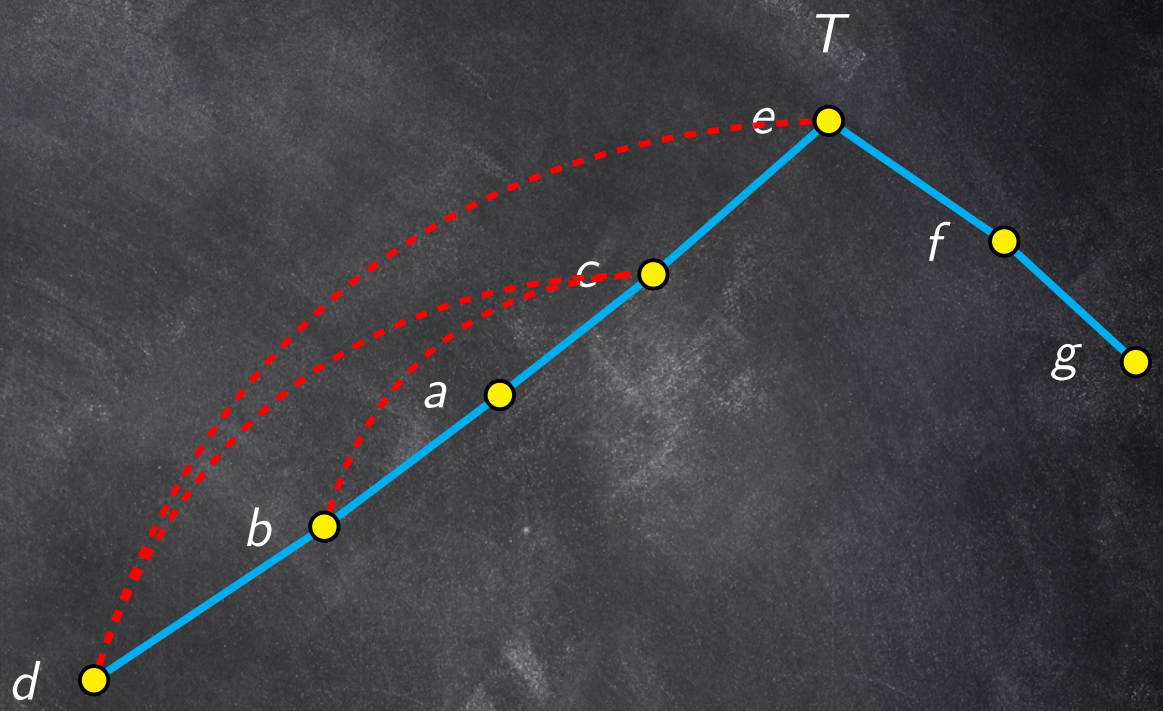
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

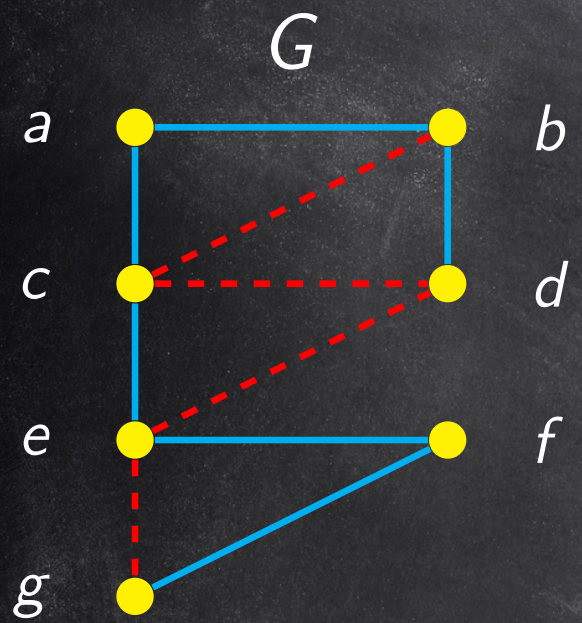
$P(g) \rightarrow N(g) = \{e, f\}$

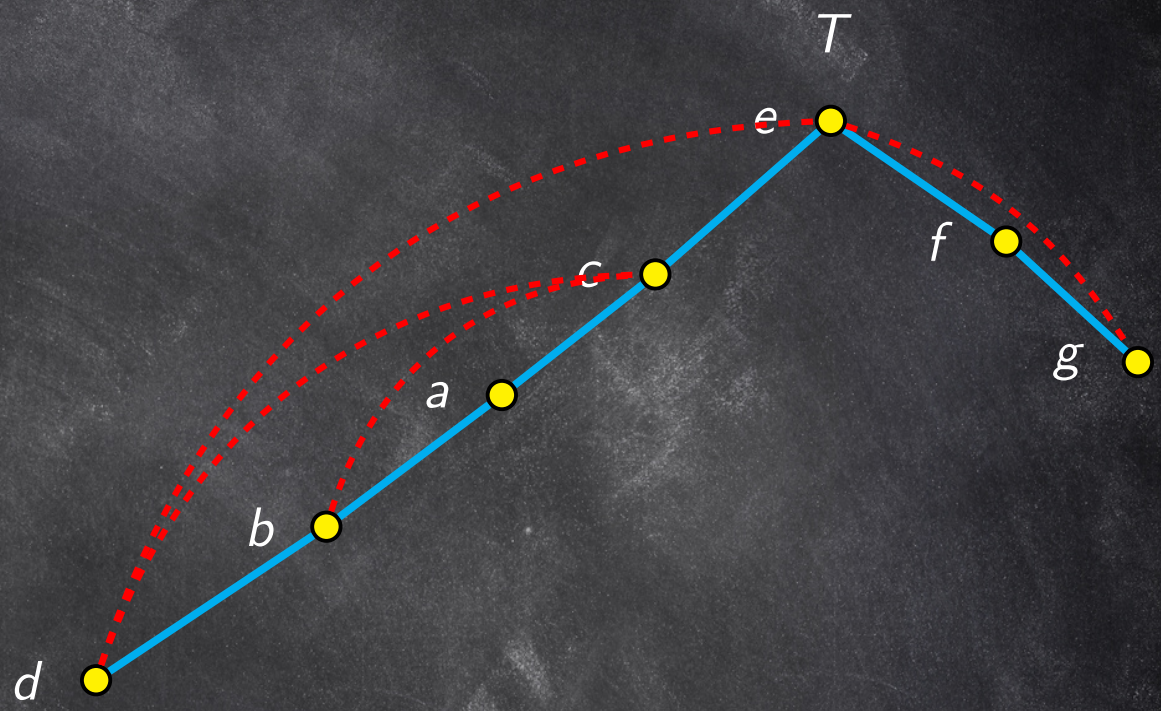
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

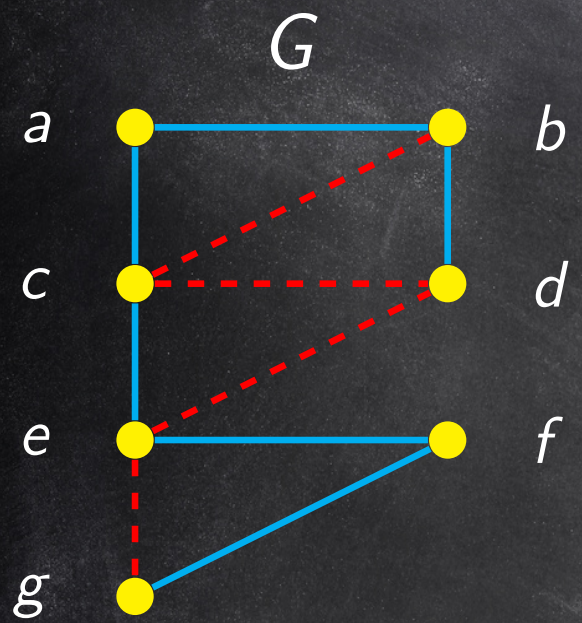
$P(g) \rightarrow N(g) = \{e, f\}$

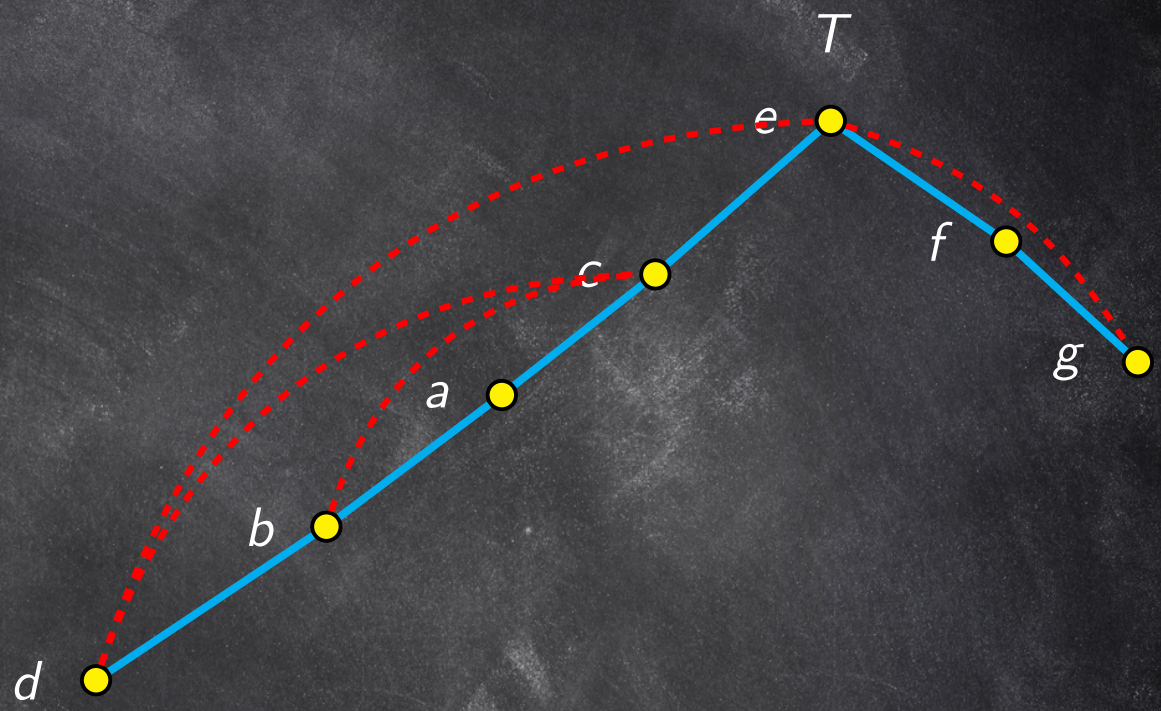
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

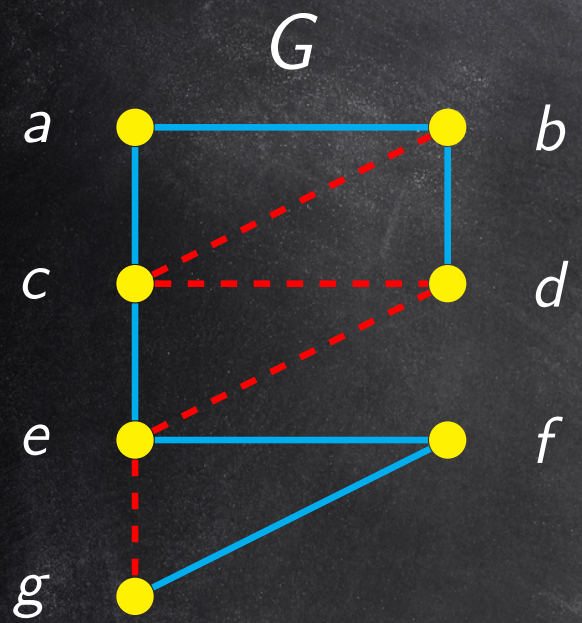
$P(g) \rightarrow N(g) = \{e, f\}$

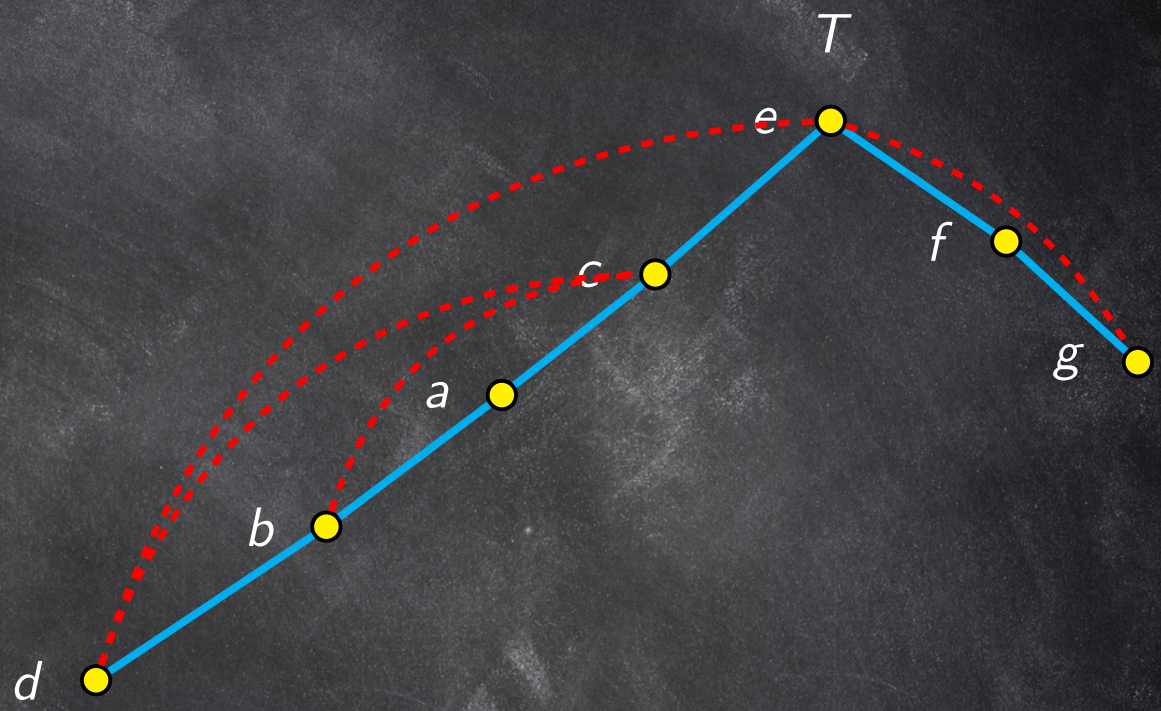
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	0

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

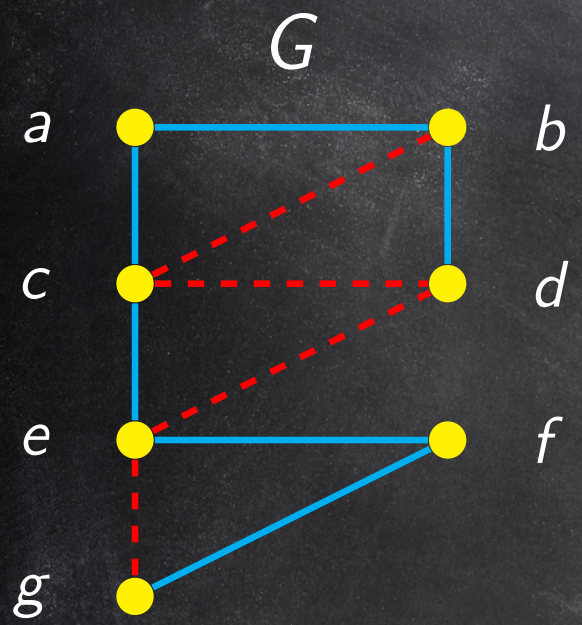
$P(g) \rightarrow N(g) = \{e, f\}$

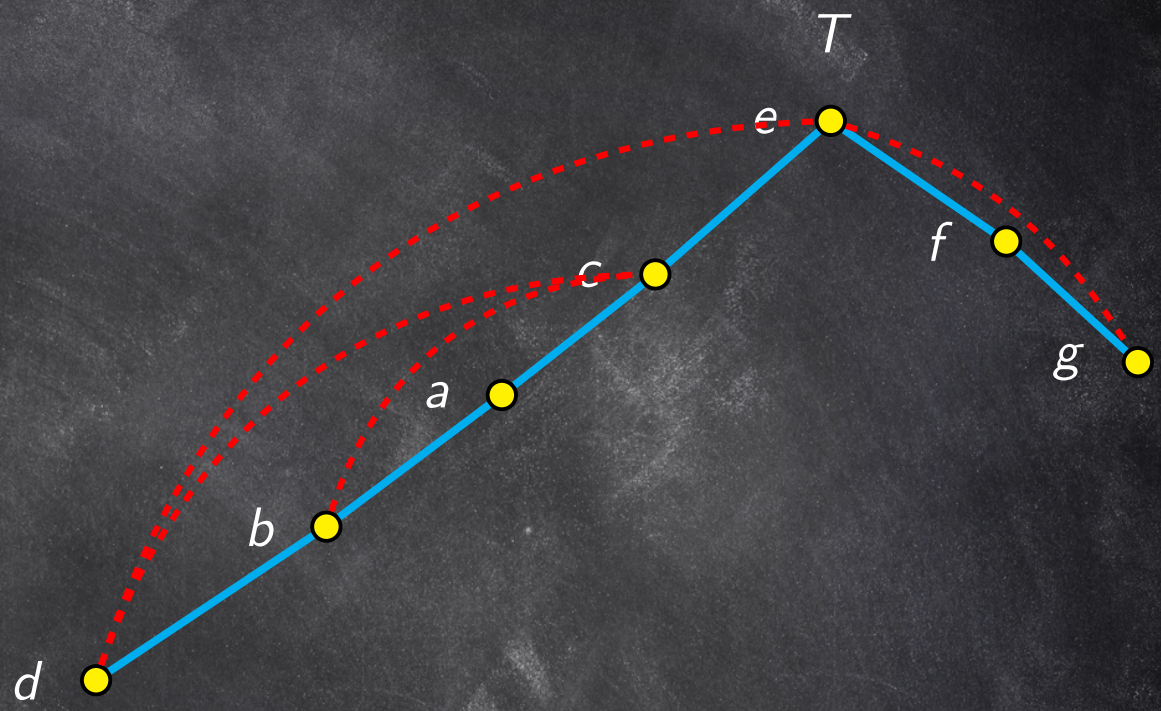
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	0	12

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

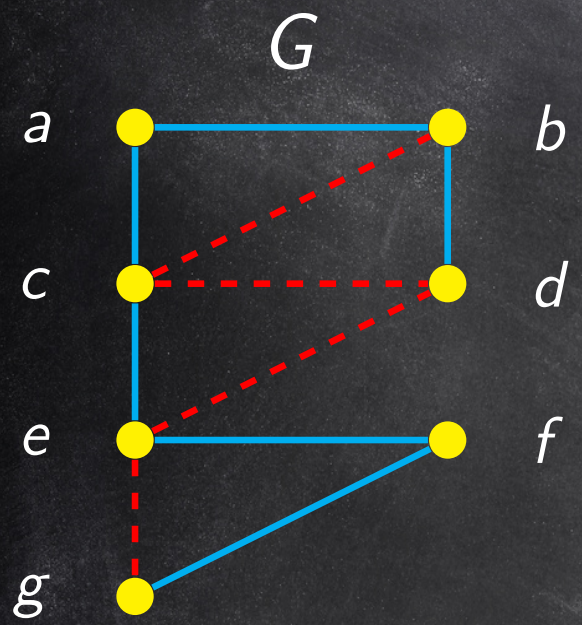
$P(g) \rightarrow N(g) = \{e, f\}$

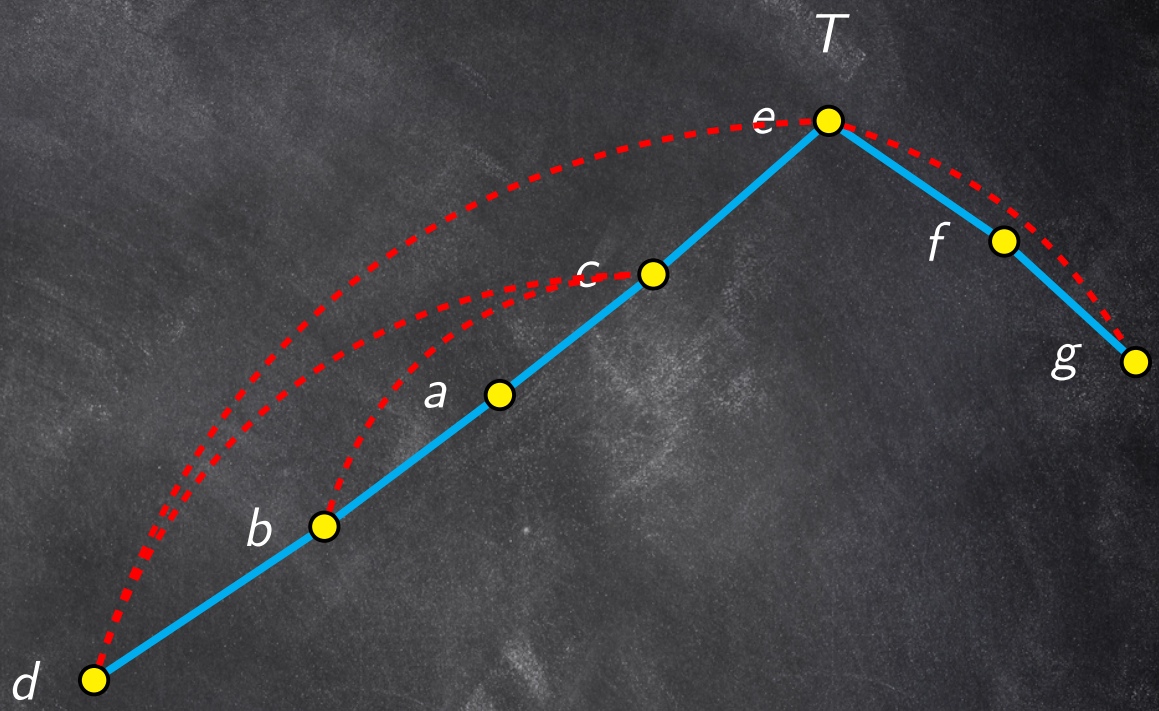
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	13	12

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

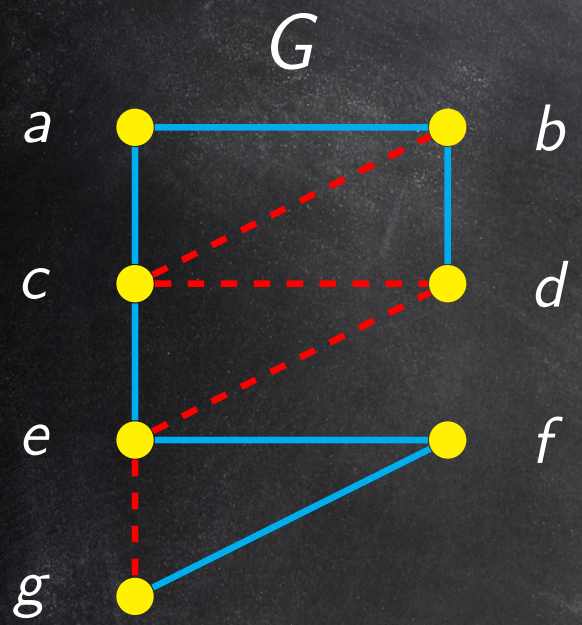
$P(g) \rightarrow N(g) = \{e, f\}$

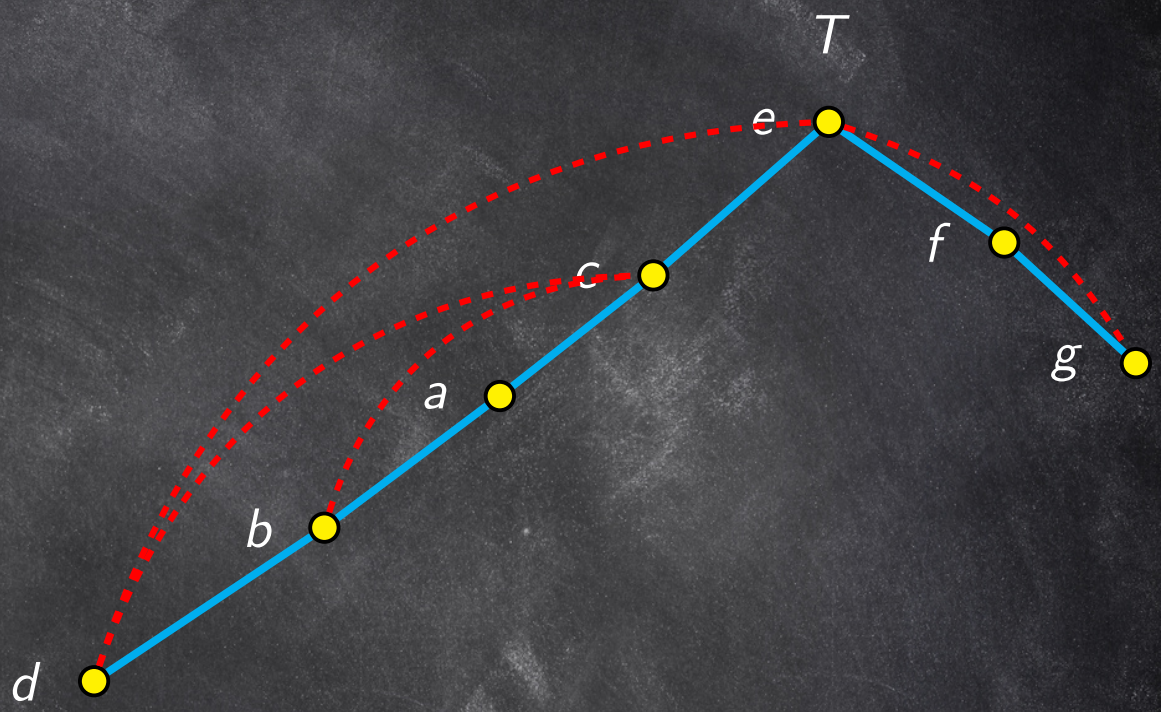
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	0	13	12

Luís Felipe

24/03/26

Execução da DFS





$P(e) \rightarrow N(e) = \{c, d, f, g\}$

$P(c) \rightarrow N(c) = \{a, b, d, e\}$

$P(a) \rightarrow N(a) = \{b, c\}$

$P(b) \rightarrow N(b) = \{a, c, d\}$

$P(d) \rightarrow N(d) = \{b, c, e\}$

$P(f) \rightarrow N(f) = \{e, g\}$

$P(g) \rightarrow N(g) = \{e, f\}$

v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	14	13	12

Luís Felipe

24/03/26

Árvore da busca

Árvore é um grafo conexo e acíclico.

Luís Felipe

24/03/26

Árvore da busca

Árvore é um grafo conexo e acíclico.

Obs.: Se o grafo de entrada G é conexo, a árvore de profundidade T é uma árvore geradora de G (isto é, uma árvore que alcança todos os vértices de G).

Luís Felipe

24/03/26

Árvore da busca

Árvore é um grafo conexo e acíclico.

Obs.: Se o grafo de entrada G é conexo, a árvore de profundidade T é uma árvore geradora de G (isto é, uma árvore que alcança todos os vértices de G).

Neste caso, todos os vértices de G são alcançados pela busca e ficam com uma PE diferente de zero no final da busca.

Luís Felipe

24/03/26

Árvore da busca

Árvore é um grafo conexo e acíclico.

Obs.: Se o grafo de entrada G é conexo, a árvore de profundidade T é uma árvore geradora de G (isto é, uma árvore que alcança todos os vértices de G).

Neste caso, todos os vértices de G são alcançados pela busca e ficam com uma PE diferente de zero no final da busca.

Somente as arestas azuis (ligando pai e filho) pertencem à árvore de profundidade T. As arestas vermelhas (arestas de retorno) não pertencem a T.

Luís Felipe

24/03/26

Busca em profundidade

Arestas de retorno

Luís Felipe

24/03/26

Busca em profundidade

Arestas de retorno

- Toda aresta de retorno fecha um ciclo.
- Toda aresta de retorno liga um vértice v a um de seus ancestrais na árvore de profundidade T.

Luís Felipe

24/03/26

Busca em profundidade

Arestas de retorno

- Toda aresta de retorno fecha um ciclo.
- Toda aresta de retorno liga um vértice v a um de seus ancestrais na árvore de profundidade T.
  - Na DFS, quando uma aresta é classificada como aresta de retorno, ela necessariamente conecta um vértice a outro que ainda está ativo na pilha de recursão. Os únicos vértices ativos nesse momento são seus ancestrais.
  - Ou seja, quando seguimos para uma nova ramificação, os vértices das ramificações anteriores já foram completamente explorados. Portanto, esses vértices não estão mais na pilha de recursão.

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

O intervalo $I(v)$ representa exatamente o período em que v permanece ativo na pilha de execução da DFS.

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

O intervalo $I(v)$ representa exatamente o período em que v permanece ativo na pilha de execução da DFS.

Durante a execução:

- Um vértice está na pilha $\iff$

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

O intervalo $I(v)$ representa exatamente o período em que v permanece ativo na pilha de execução da DFS.

Durante a execução:

- Um vértice está na pilha $\iff$ $PE(v) > 0$ e $PS(v) = 0$;

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

O intervalo $I(v)$ representa exatamente o período em que v permanece ativo na pilha de execução da DFS.

Durante a execução:

- Um vértice está na pilha $\iff$ $PE(v) > 0$ e $PS(v) = 0$;
- Os vértices ativos formam sempre um caminho da raiz até o vértice corrente.

Luís Felipe

24/03/26

Intervalos de vida e pilha de execução na DFS

Definição (Intervalo de vida).
Para cada vértice v, definimos:

$$I(v) = [PE(v),\ PS(v)]$$

onde:

- $PE(v)$ é o tempo de descoberta (entrada);
- $PS(v)$ é o tempo de finalização (saída).

O intervalo $I(v)$ representa exatamente o período em que v permanece ativo na pilha de execução da DFS.

Durante a execução:

- Um vértice está na pilha $\iff$ $PE(v) > 0$ e $PS(v) = 0$;
- Os vértices ativos formam sempre um caminho da raiz até o vértice corrente.

Caracterização de ancestralidade.
Para vértices v e w: v é descendente de w $\iff$ $I(v) \subset I(w)$
ou equivalentemente: $PE(w) < PE(v)$ e $PS(v) < PS(w)$.

Luís Felipe

24/03/26

Busca em profundidade

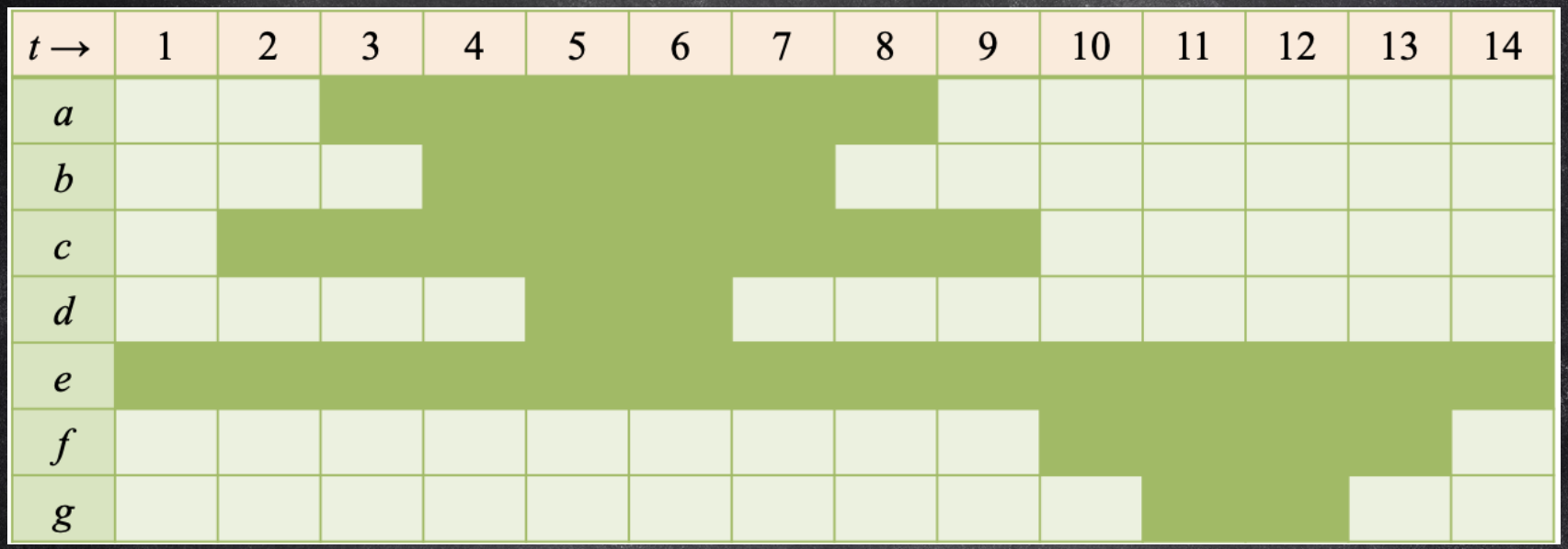
v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	14	13	12

Luís Felipe

24/03/26

Busca em profundidade

v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	14	13	12

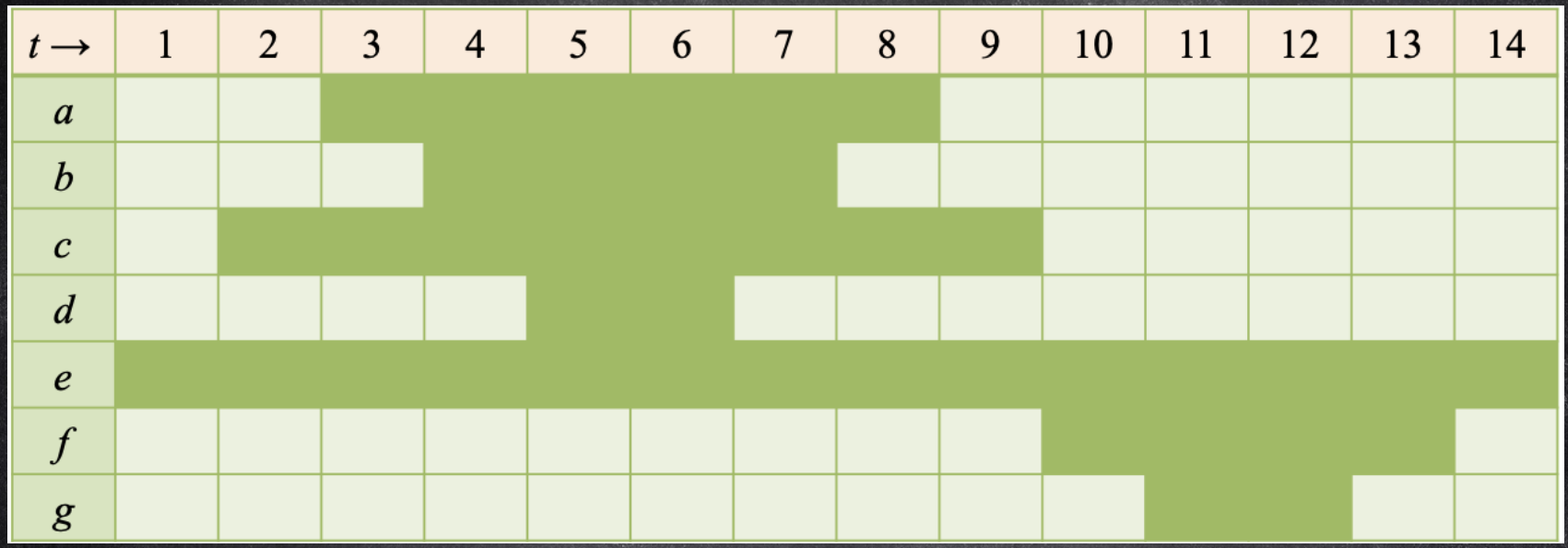


Luís Felipe

24/03/26

Busca em profundidade

v	a	b	c	d	e	f	g
PE(v)	3	4	2	5	1	10	11
PS(v)	8	7	9	6	14	13	12



Intervalos de vida dos vértices: v é descendente de w na árvore de profundidade T se e somente se o intervalo de vida de v está contido no intervalo de vida de w. Isto é: $PE(v) > PE(w)$ e $PS(v) < PS(w)$ (v "entra depois" e "sai antes" de w).

Luís Felipe

24/03/26

Aplicação 1: Conexidade e número de componentes

- Problema: Dado um grafo G, verificar se G é conexo.

Luís Felipe

24/03/26

Aplicação 1: Conexidade e número de componentes

- Problema: Dado um grafo G, verificar se G é conexo.
- Ideia: rodar DFS repetidamente enquanto existir vértice não descoberto.

Luís Felipe

24/03/26

Aplicação 1: Conexidade e número de componentes

- Problema: Dado um grafo G, verificar se G é conexo.
- Ideia: rodar DFS repetidamente enquanto existir vértice não descoberto.
- Resultado: o contador c ao final é o número de componentes conexas.

Luís Felipe

24/03/26

Aplicação 1: Conexidade e número de componentes

- Problema: Dado um grafo G, verificar se G é conexo.
- Ideia: rodar DFS repetidamente enquanto existir vértice não descoberto.
- Resultado: o contador c ao final é o número de componentes conexas.

```
1  c <- 0
2  enquanto existe v em V(G) tal que PE(v) = 0 faca
3      c <- c + 1
4      executar P(v)                  // P(v) é o procedimento DFS com raiz v
```

Luís Felipe

24/03/26

Aplicação 1: Conexidade e número de componentes

- Problema: Dado um grafo G, verificar se G é conexo.
- Ideia: rodar DFS repetidamente enquanto existir vértice não descoberto.
- Resultado: o contador c ao final é o número de componentes conexas.

```
1  c <- 0
2  enquanto existe v em V(G) tal que PE(v) = 0 faca
3      c <- c + 1
4      executar P(v)                  // P(v) é o procedimento DFS com raiz v
```

- Se $c = 1$, então G é conexo.
- Se $c > 1$, então G é desconexo.

Luis Felipe

24/03/26

```python
def dfs(v, adj, pe):
    pe[v] = 1               # Marca o vertice v como descoberto

    # Percorre todos os vizinhos de v
    for w in adj[v]:
        if pe[w] == 0:
            dfs(w, adj, pe)

def numero_componentes(adj):
    n = len(adj)
    pe = [0] * n            # PE(v) = 0 significa "nao descoberto"

    c = 0
    for v in range(n):
        if pe[v] == 0:
            c += 1
            dfs(v, adj, pe)
    return c

def eh_conexo(adj):
    return numero_componentes(adj) == 1

# Exemplo
if __name__ == "__main__":
    # grafo com 5 vertices (0..4)
    # 0-1-2 eh um componente; 3-4 eh outro => c = 2
    adj = [
        [1],        # 0
        [0, 2],     # 1
        [1],        # 2
        [4],        # 3
        [3],        # 4
    ]

    c = numero_componentes(adj)
    print("Numero de componentes:", c)

    if c == 1:
        print("Grafo conexo")
    else:
        print("Grafo desconexo")
```

Luís Felipe

24/03/26

Aplicação 2: Existe caminho de v até w?

- Problema: Dado G e vértices v, w, decidir se existe caminho de v a w.

Luís Felipe

24/03/26

Aplicação 2: Existe caminho de v até w?

- Problema: Dado G e vértices v, w, decidir se existe caminho de v a w.
- Solução: executar uma única DFS com raiz v.
- Se ao final $PE(w) = 0$, então w não foi alcançado. Logo não existe caminho.
- Caso contrário, se $PE(w) \neq 0$, então w foi alcançado, e existe um caminho de v a w em G; neste caso, w será descendente de v na árvore de profundidade T, e para determinar o caminho basta rodar o algoritmo a seguir.

Luís Felipe

24/03/26

Aplicação 2: Existe caminho de v até w?

- Problema: Dado G e vértices v, w, decidir se existe caminho de v a w.
- Solução: executar uma única DFS com raiz v.
- Se ao final $PE(w) = 0$, então w não foi alcançado. Logo não existe caminho.
- Caso contrário, se $PE(w) \neq 0$, então w foi alcançado, e existe um caminho de v a w em G; neste caso, w será descendente de v na árvore de profundidade T, e para determinar o caminho basta rodar o algoritmo a seguir.

Reconstruindo um caminho (usando *pai*)

Luís Felipe

24/03/26

Aplicação 2: Existe caminho de v até w?

- Problema: Dado G e vértices v, w, decidir se existe caminho de v a w.
- Solução: executar uma única DFS com raiz v.
- Se ao final $PE(w) = 0$, então w não foi alcançado. Logo não existe caminho.
- Caso contrário, se $PE(w) \neq 0$, então w foi alcançado, e existe um caminho de v a w em G; neste caso, w será descendente de v na árvore de profundidade T, e para determinar o caminho basta rodar o algoritmo a seguir.

Reconstruindo um caminho (usando *pai*)
Se $PE(w) \neq 0$, então existe caminho; para extrair um caminho:

```
1  x <- w
2  C <- x
3  enquanto x != v faca
4      x <- pai(x)
5      colocar x a esquerda em C
6  imprimir C                  // Obs: C nao é necessariamente o melhor
```

Luis Felipe
24/03/26

```python
def dfs_com_pai(v, adj, pe, pai):
    pe[v] = 1
    for w in adj[v]:
        if pe[w] == 0:
            pai[w] = v
            dfs_com_pai(w, adj, pe, pai)

def caminho_v_ate_w(adj, v, w):
    n = len(adj)
    pe = [0] * n                    # PE(u)=0: nao descoberto; PE(u)=1: descoberto
    pai = [-1] * n                  # pai[u] = predecessor na arvore DFS (-1 = null)

    dfs_com_pai(v, adj, pe, pai)    # Executa DFS a partir de v (raiz)

    if pe[w] == 0:                  # Se w nao foi alcancado, nao existe caminho
        return None

    # Reconstrucao do caminho usando pai
    x = w
    C = [x]
    while x != v:
        x = pai[x]
        C.insert(0, x)              # coloca a esquerda (inicio da lista)
    return C

if __name__ == "__main__":
    adj = [
        [1, 2],       # 0
        [0, 3],       # 1
        [0, 3, 4],    # 2
        [1, 2, 5],    # 3
        [2],          # 4
        [3, 6],       # 5
        [5]           # 6
    ]
    v = 0
    w = 6
    C = caminho_v_ate_w(adj, v, w)
    if C is None:
        print("Nao existe caminho de", v, "ate", w)
    else:
        print("Existe caminho de", v, "ate", w)
        print("Um caminho encontrado:", C)
```

Luís Felipe

24/03/26

Aplicação 3: Labirinto

- Problema: Dado um labirinto, determinar um caminho da entrada até a saída (se existir).

Luís Felipe

24/03/26

Aplicação 3: Labirinto

- Problema: Dado um labirinto, determinar um caminho da entrada até a saída (se existir).
- Modelagem: cada célula/canto navegável vira um vértice; conexões viram arestas.

Luís Felipe

24/03/26

Aplicação 3: Labirinto

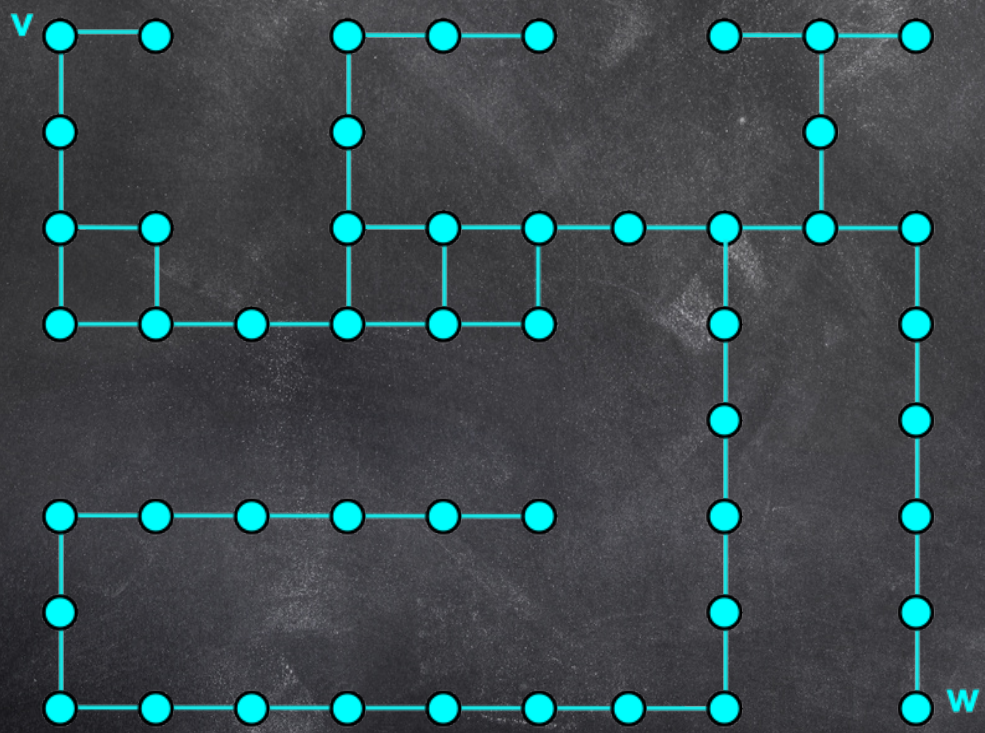
- Problema: Dado um labirinto, determinar um caminho da entrada até a saída (se existir).
- Modelagem: cada célula/canto navegável vira um vértice; conexões viram arestas.
- Solução: reduzir para a Aplicação 2 (caminho entre dois vértices).

Luís Felipe

24/03/26

Aplicação 3: Solução (redução para grafo)

- Montar o grafo correspondente ao labirinto.
- Executar DFS a partir da entrada v.
- Se a saída w for alcançada, reconstruir o caminho usando $pai(\cdot)$.



Luís Felipe

24/03/26

Aplicação 4: Encontrar ciclo ou concluir que é acíclico

- Problema: Dado G, encontrar um ciclo em G ou concluir que G é acíclico.

Luís Felipe

24/03/26

Aplicação 4: Encontrar ciclo ou concluir que é acíclico

- Problema: Dado G, encontrar um ciclo em G ou concluir que G é acíclico.
- Solução: executar DFS.

Luís Felipe

24/03/26

Aplicação 4: Encontrar ciclo ou concluir que é acíclico

- Problema: Dado G, encontrar um ciclo em G ou concluir que G é acíclico.
- Solução: executar DFS. Teremos dois casos:
  - Se não aparece nenhuma aresta de retorno, então G é acíclico.
  - Se aparece uma aresta de retorno vw, então ela fecha um ciclo. Neste caso, v é descendente de w na árvore, e um ciclo C é formado. Para determinar C, basta rodar o algoritmo da Aplicação 2.

Luís Felipe

24/03/26

Aplicação 5: Decidir se o grafo é 2-colorível (bipartido)

- G é 2-colorível se podemos colorir os vértices com duas cores de modo que vértices vizinhos não tenham a mesma cor.

Luís Felipe

24/03/26

Aplicação 5: Decidir se o grafo é 2-colorível (bipartido)

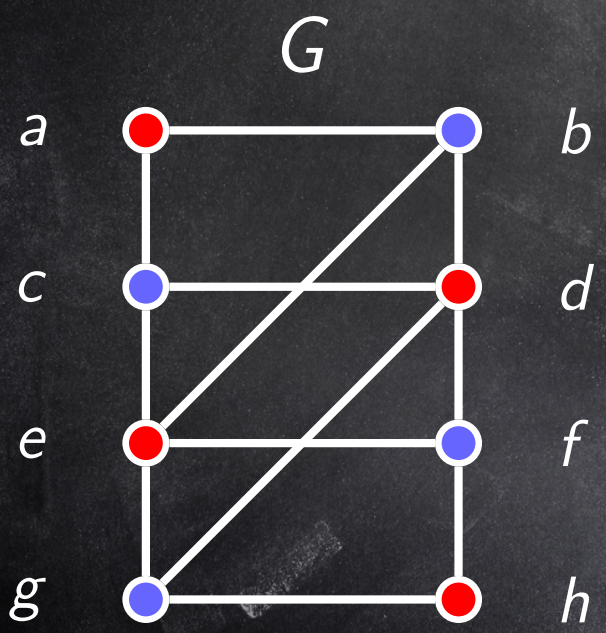
- G é 2-colorível se podemos colorir os vértices com duas cores de modo que vértices vizinhos não tenham a mesma cor.
- Teorema: G é 2-colorível sse G não contém ciclo ímpar (Aula 3).

Luís Felipe

24/03/26

Aplicação 5: Decidir se o grafo é 2-colorível (bipartido)

- G é 2-colorível se podemos colorir os vértices com duas cores de modo que vértices vizinhos não tenham a mesma cor.
- Teorema: G é 2-colorível sse G não contém ciclo ímpar (Aula 3).



G é 2-colorível

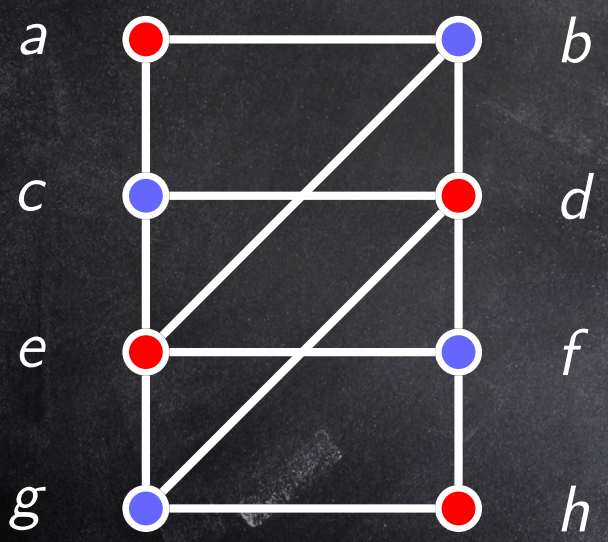
Luís Felipe

24/03/26

Aplicação 5: Decidir se o grafo é 2-colorível (bipartido)

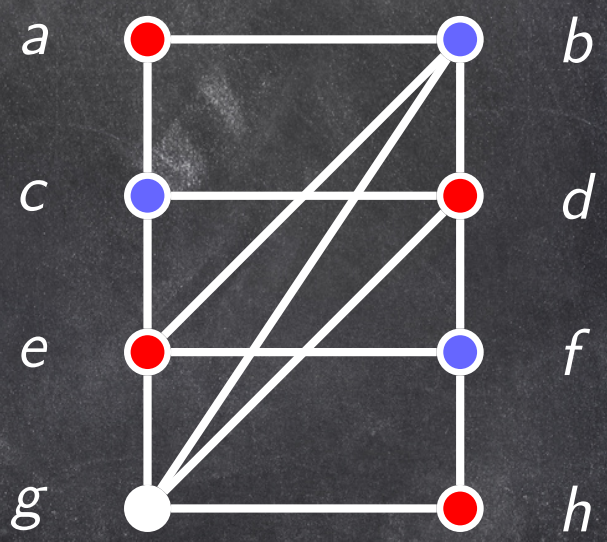
- G é 2-colorível se podemos colorir os vértices com duas cores de modo que vértices vizinhos não tenham a mesma cor.
- Teorema: G é 2-colorível sse G não contém ciclo ímpar (Aula 3).

G



G é 2-colorível

H



H não é 2-colorível

Luís Felipe

24/03/26

Aplicação 5: 2-coloribilidade (Bipartido)

```python
def dfs_color(v, adj, cor, pai):
    """
    Tenta colorir a componente de v com duas cores (0/1).
    Retorna False se encontrar conflito (ciclo impar), True caso contrario.
    """
    for w in adj[v]:
        if cor[w] == -1:
            pai[w] = v
            cor[w] = 1 - cor[v]
            if not dfs_color(w, adj, cor, pai):
                return False
        elif w != pai[v]:
            if cor[w] == cor[v]:
                return False
    return True

def eh_2colorivel(adj):
    """
    Retorna True se o grafo (possivelmente desconexo) for 2-colorivel.
    adj: lista de adjacencia.
    """
    n = len(adj)
    cor = [-1] * n      # -1 = sem cor, 0 e 1 sao as duas cores
    pai = [-1] * n

    for s in range(n):
        if cor[s] == -1:
            cor[s] = 0
            pai[s] = -1
            if not dfs_color(s, adj, cor, pai):
                return False
    return True
```

Luís Felipe

24/03/26

Aplicação 5: Exemplo

```
1  if __name__ == "__main__":
2      # Exemplo 1: bipartido (um quadrado 0-1-2-3-0)
3      adj1 = [
4          [1, 3],  # 0
5          [0, 2],  # 1
6          [1, 3],  # 2
7          [2, 0],  # 3
8      ]
9      print("G1 2-colorivel?", eh_2colorivel(adj1))
10
11     # Exemplo 2: nao bipartido (triangulo 0-1-2-0)
12     adj2 = [
13         [1, 2],  # 0
14         [0, 2],  # 1
15         [0, 1],  # 2
16     ]
17     print("G2 2-colorivel?", eh_2colorivel(adj2))
```