Melhores momentos

AULA 1

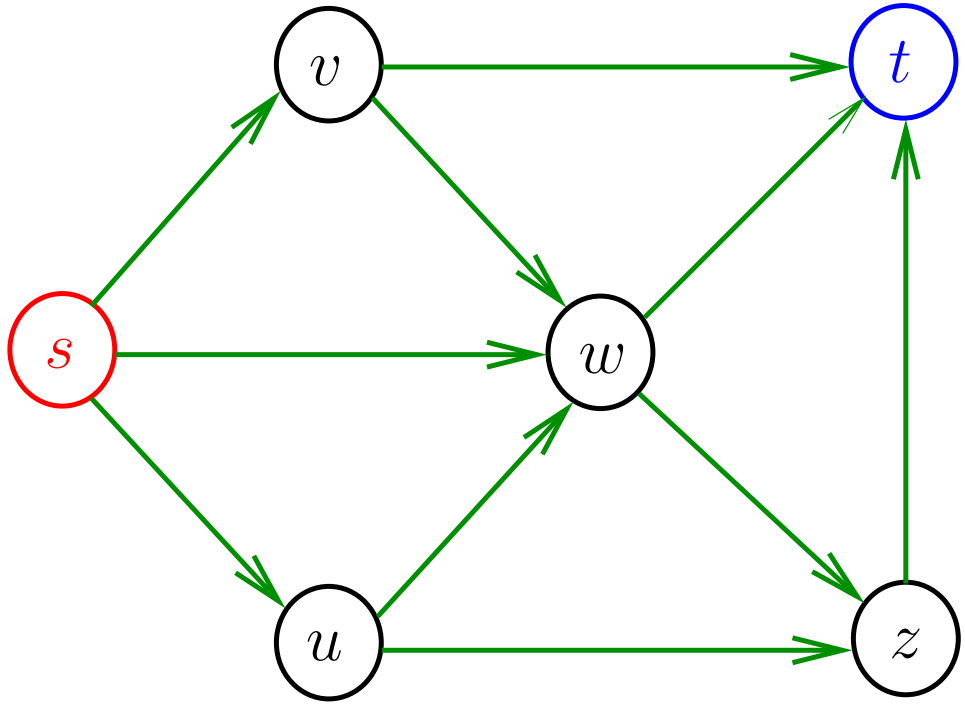
Busca

Problema de busca: Dados nós s e t de um grafo, encontrar um caminho de s a t

Busca

Problema de busca: Dados nós s e t de um grafo, encontrar um caminho de s a t

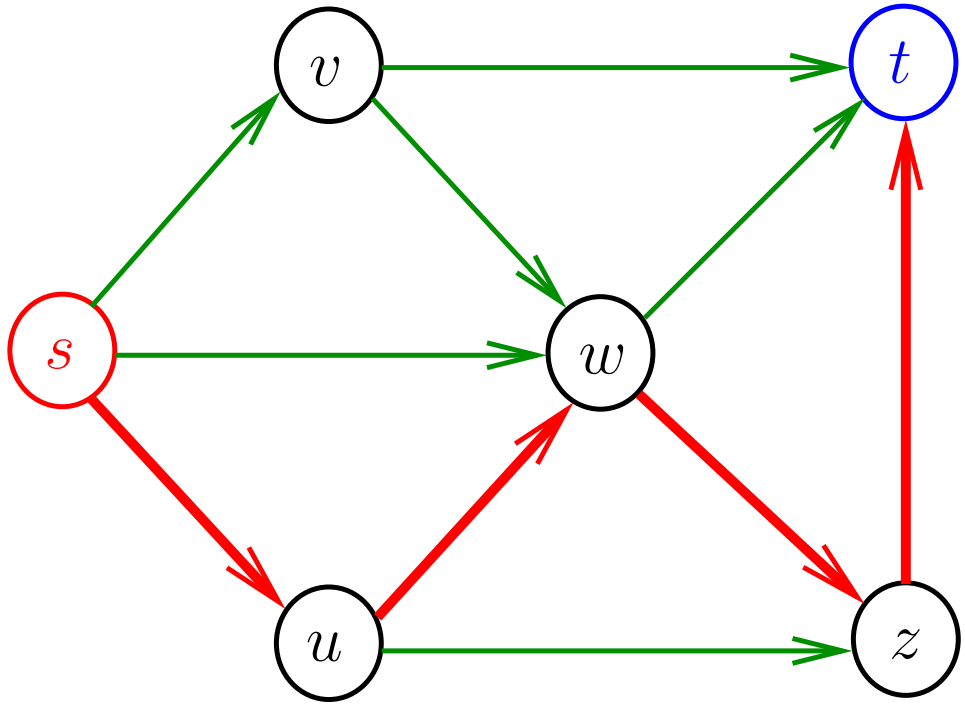
Entra:



Busca

Problema de busca: Dados nós s e t de um grafo, encontrar um caminho de s a t

Sai:



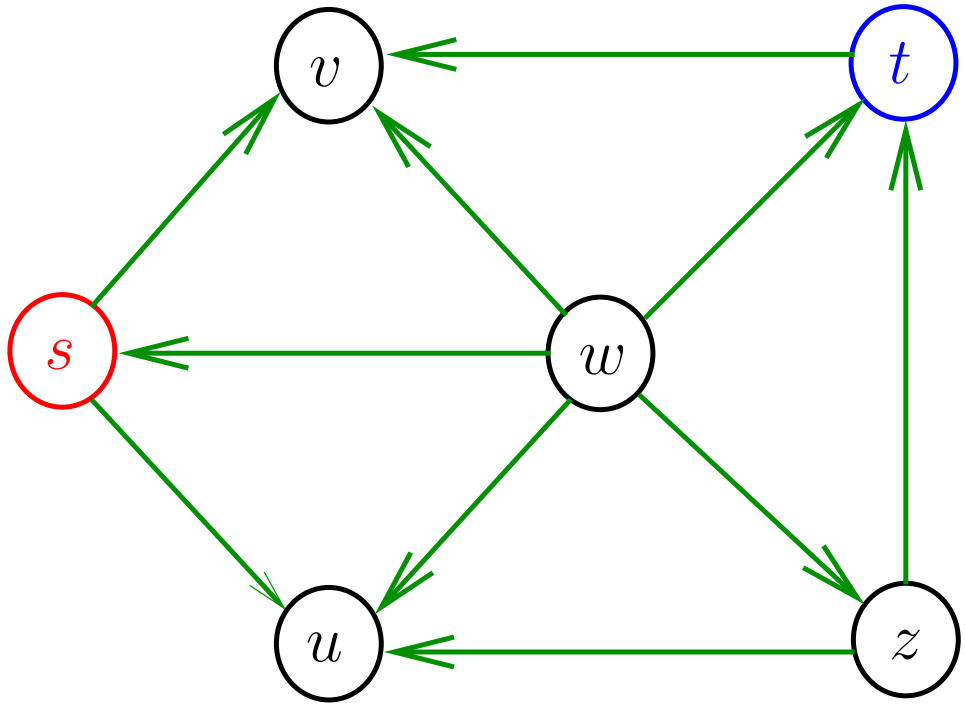
Condição de inexistência

Como é possível demonstrar que o problema **não** tem solução?

Condição de inexistência

Como é possível demonstrar que o problema **não** tem solução?

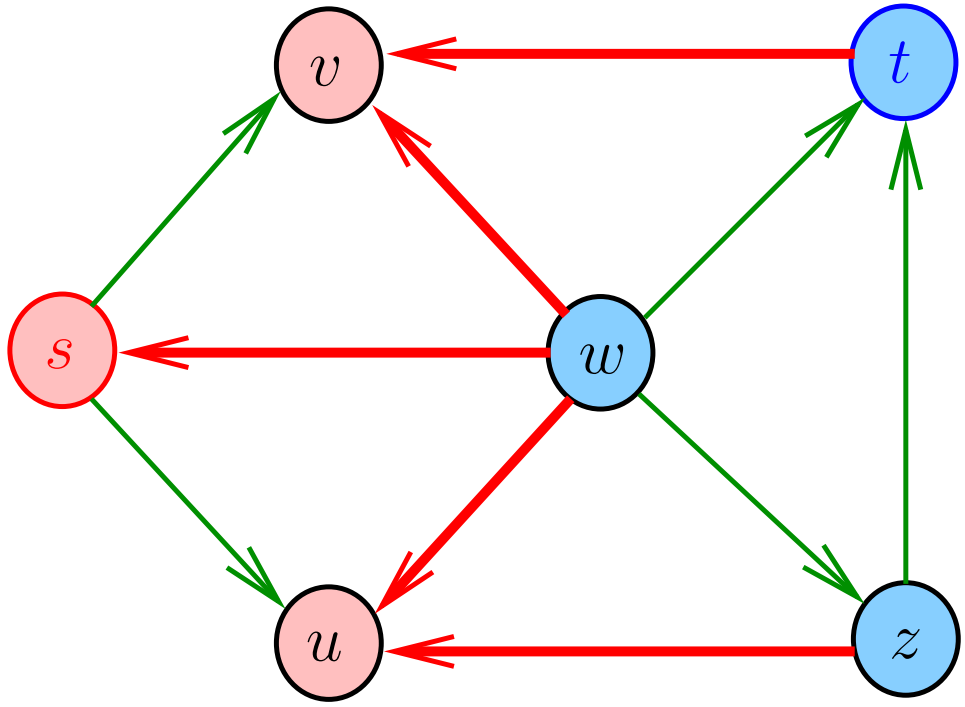
Entra:



Condição de inexistência

Como é possível demonstrar que o problema **não** tem solução?

Sai: um “certo corte” separando s de t

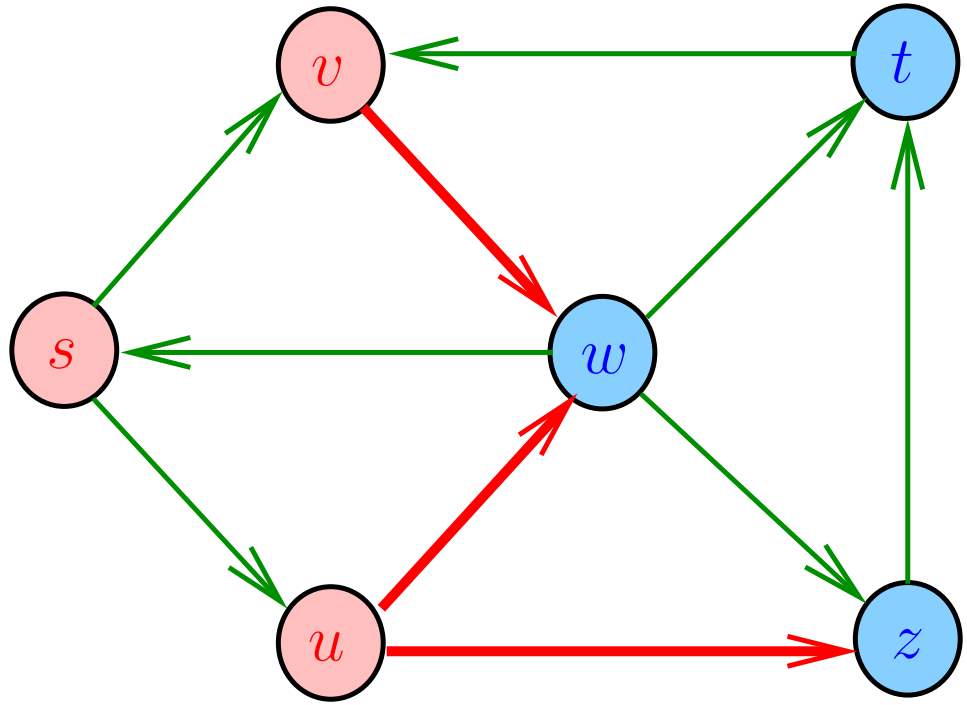


Cortes

Para quaisquer partes S e T de N:

$$\nabla(S,T) := \{st \in A : s \in S \text{ e } t \in T\}.$$

Exemplo: $\nabla(\{s,v,u\},\{w,t,z\}) = \nabla(\{v,u\},\{w,t,z\})$ são os arcos em vermelho.

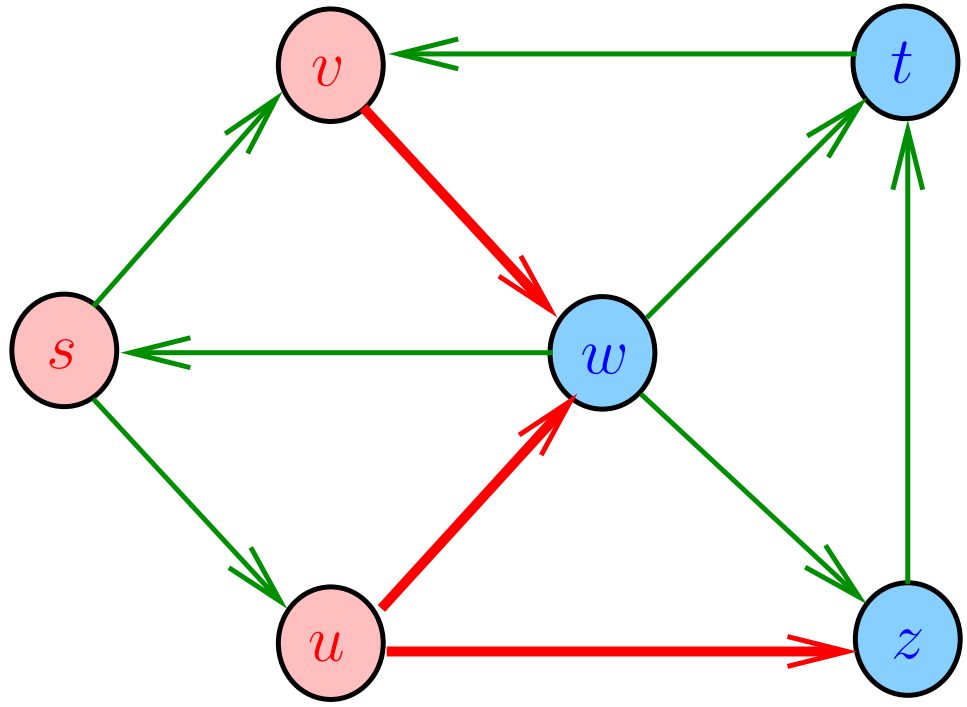


(s,t)-Cortes

Para s em $N-T$ e t em T dizemos que:

- T separa s de t
- $\nabla(N-T,T)$ separa s de t
- $\nabla(N-T,T)$ é um (s,t)-corte.

Exemplo: $\{w,t,z\}$ separa s de t

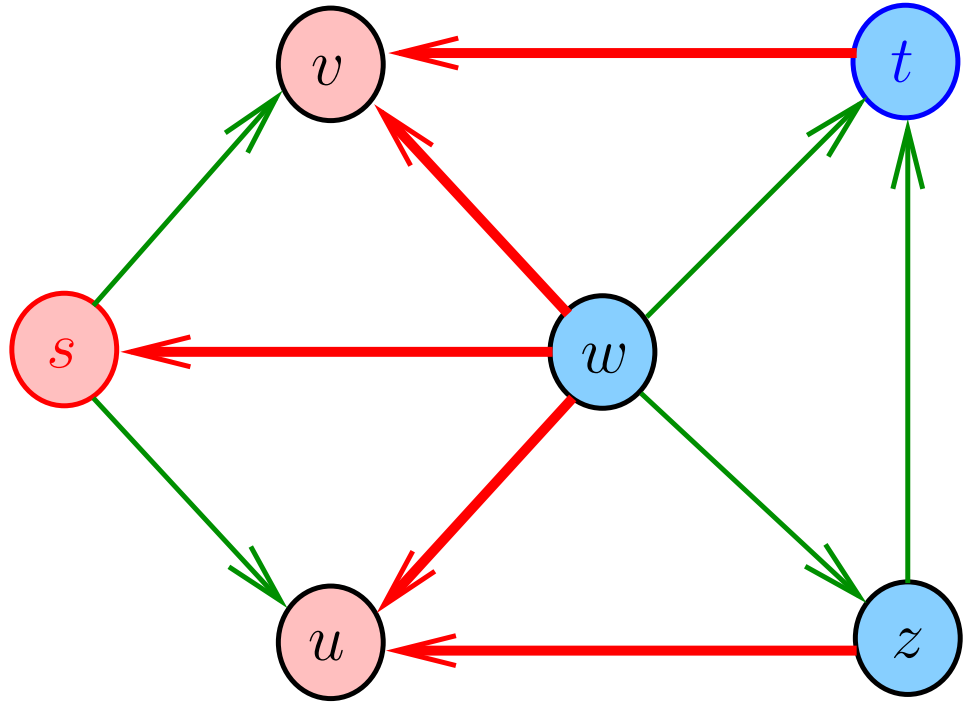


Condição de inexistência

Para demonstrar que uma dada instância não tem solução basta exibir um (s, t)-corte tal que

$$\nabla(N - T, T) = \emptyset,$$

Exemplo: um (s, t)-corte vazio

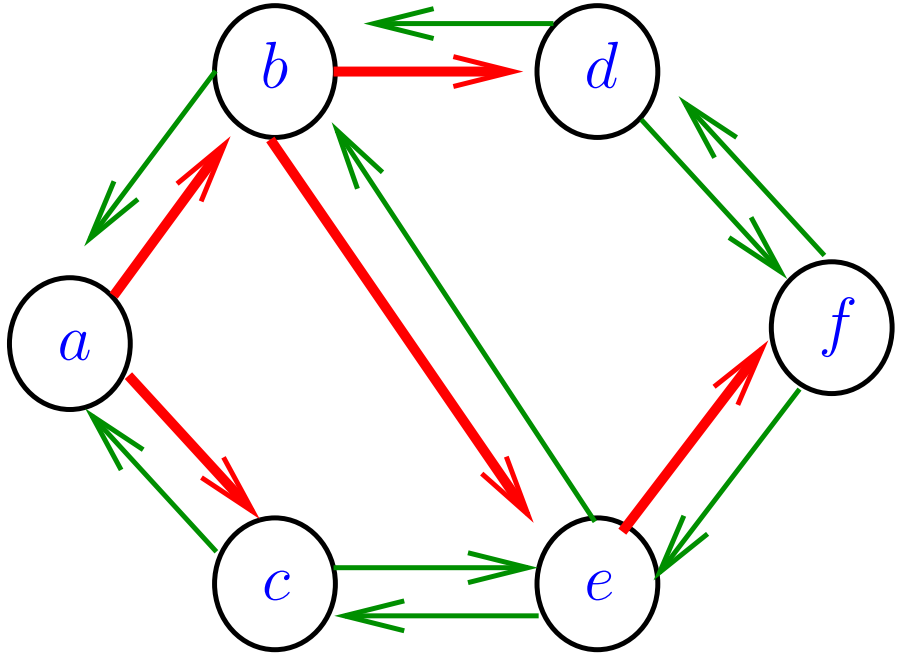


Caminhos no computador

Uma maneira compacta de representar caminhos de um nó aos nós de um grafo é através de função-predecessor.

Uma função-predecessor é uma função "parcial" π de N em N tal que, para cada j em N ,

$$\pi(j) = \text{NIL} \quad \text{ou} \quad (\pi(j), j) \in A$$

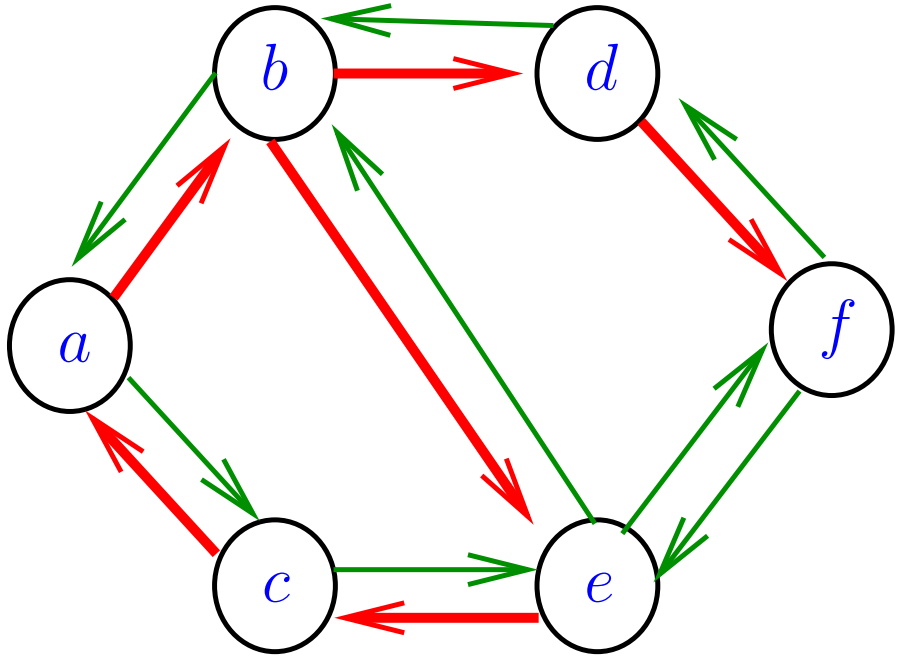


nó	π
a	NIL
b	a
c	a
d	b
e	b
f	e

Grafo de predecessores

Suponha que π é uma função-predecessor.
Denotaremos por A_π os arcos da forma $(\pi(j), j)$.
Diremos que (N, A_π) é o grafo de predecessores.

Exemplo:



nó	π
a	c
b	a
c	e
d	b
e	b
f	d

AULA 2

Problema da Busca (continuação)

PF 3.2, 3.3, 3.4

Busca genérico

Recebe dois nós s e t de um grafo (N, A) e devolve uma caminho de s a t ou um (s,t)-corte vazio.

BUSCA-GENÉRICO (N, A, s, t)

0 **para cada** i em N **faça**
1 $\quad T \leftarrow T \cup \{i\}$
2 $\quad \pi(i) \leftarrow \text{NIL}$
3 $S \leftarrow \{s\} \quad T \leftarrow T - \{s\}$

4 **enquanto** existe $ij \in A$ com $i \in S$ e $j \in T$ **faça**
5 $\quad S \leftarrow S \cup \{j\} \quad T \leftarrow T - \{j\}$
6 $\quad \pi(j) \leftarrow i$

7 **se** $t \in S$
8 $\quad$ **então devolva** o st-caminho no grafo (N, A_π)
9 $\quad$ **senão devolva** S e $T \quad \triangleright \nabla(S,T) = \emptyset$

Busca genérico

BUSCA-GENÉRICO (N, A, s, t). Recebe um grafo (N, A), e dois nós s e t e devolve st-caminho ou $T \subset N$ que separa s de t e $\nabla(N - T, T) = \emptyset$.

Cada iteração começa com uma partição (S, T) de N e uma função-predecessor π. No início da primeira iteração $S = \{s\}$, $T = N - \{s\}$ e $\pi(i) = \text{NIL}$ para todo nó i. Cada iteração consiste no seguinte.

Busca genérico

BUSCA-GENÉRICO (N, A, s, t). Recebe um grafo (N, A), e dois nós s e t e devolve st-caminho ou $T \subset N$ que separa s de t e $\nabla(N - T, T) = \emptyset$.

Cada iteração começa com uma partição (S, T) de N e uma função-predecessor π. No início da primeira iteração $S = \{s\}$, $T = N - \{s\}$ e $\pi(i) = \text{NIL}$ para todo nó i. Cada iteração consiste no seguinte.

Caso 1: não existe $ij \in A$ com $i \in S$ e $j \in T$

Caso 1A: $t \in S$

Devolva o st-caminho no grafo (N, A_π) e pare.

Caso 1B: $t \notin S$

Devolva T e pare.

Busca genérico

BUSCA-GENÉRICO (N, A, s, t). Recebe um grafo (N, A), e dois nós s e t e devolve st-caminho ou $T \subset N$ que separa s de t e $\nabla(N - T, T) = \emptyset$.

Cada iteração começa com uma partição (S, T) de N e uma função-predecessor π. No início da primeira iteração $S = \{s\}$, $T = N - \{s\}$ e $\pi(i) = \text{NIL}$ para todo nó i. Cada iteração consiste no seguinte.

Caso 2: existe $ij \in A$ com $i \in S$ e $j \in T$

$S' \leftarrow S \cup \{j\}$

$T' \leftarrow T - \{j\}$

Seja π' tal que $\pi'(k) := \begin{cases} \pi(k) & \text{se } k \neq j, \\ i & \text{se } k = j. \end{cases}$

Comece nova iteração com S', T' e π' nos papéis de S, T e π.

Invariantes

Na linha 4, antes da verificação da condição "existe $ij \in A \ldots$" valem as seguintes invariantes:

(i0) S e T formam uma partição de N ou seja,

$$S \cap T = \emptyset \text{ e } S \cup T = N$$

(i1) para cada arco pq no grafo de predecessores tem-se $p \in S$ e $q \in S$;

(i2) $\pi(s) = \text{NIL}$ e $s \in S$;

(i3) para cada nó v distinto de s,

$v \in S \Leftrightarrow \pi(v) \neq \text{NIL}$;

(i4) para cada nó v, se $\pi(v) \neq \text{NIL}$ então existe um caminho de s a v no grafo de predecessores.

Correção

No início da última iteração:

- $\nabla(S, T) = \emptyset$
- se $t \in S$ então $\pi(t) \neq \text{NIL}$ (por (i3)), logo existe caminho de s a t (por (i4))
- se $t \notin S$ então $\nabla(S, T)$ é um (s, t)-corte (por (i2) e (i0)) vazio

Conclusão: o algoritmo faz o que promete.

Conclusão

Para quaisquer nós s e t de um grafo (N, A), vale uma e apenas umas das seguintes afirmações:

- existe um caminho de s a t
- existe (s, t)-corte vazio.

Condição de inexistências, ainda . . .

Um 0-potencial é qualquer função y de N em $\{0, 1\}$ ($\mathbb{Z}$) tal que

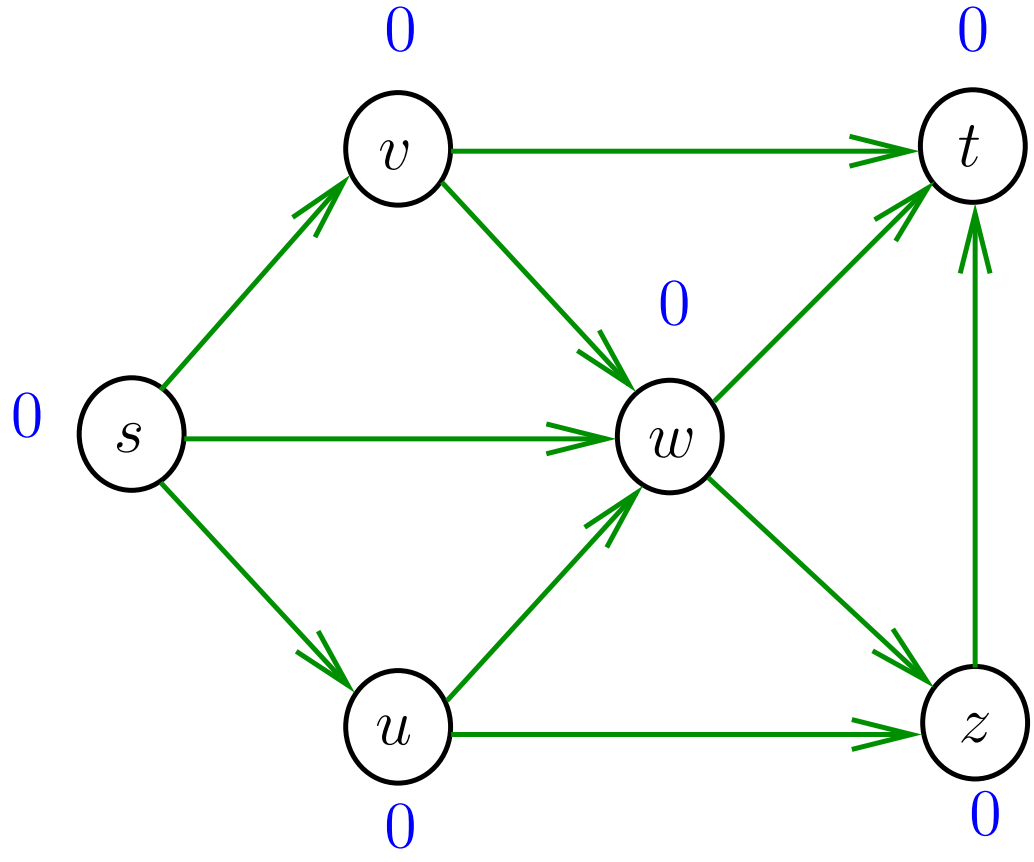
$$y(j) - y(i) \leq 0 \quad \text{para todo arco } ij.$$

Condição de inexistências, ainda . . .

Um 0-potencial é qualquer função y de N em $\{0, 1\}$ ($\mathbb{Z}$) tal que

$$y(j) - y(i) \leq 0 \quad \text{para todo arco } ij.$$

Exemplo 1: 0-potencial constante (sem graça. . .)

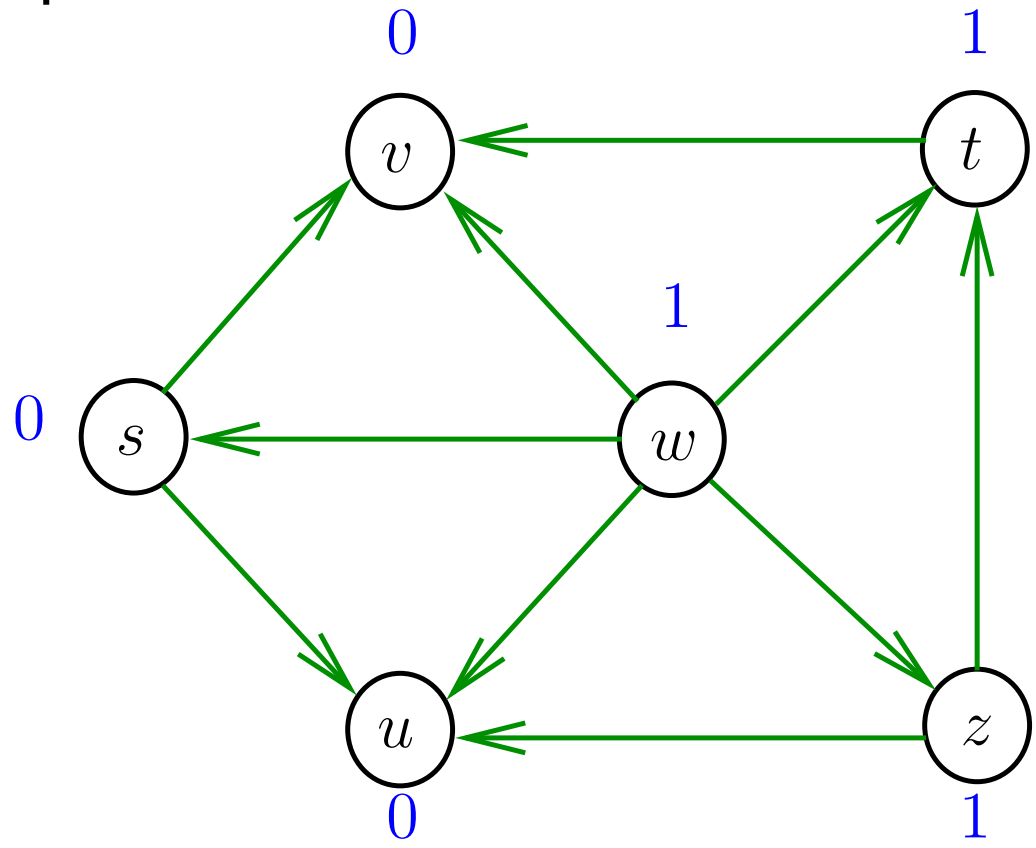


Condição de inexistências, ainda . . .

Um 0-potencial é qualquer função y de N em $\{0, 1\}$ ($\mathbb{Z}$) tal que

$$y(j) - y(i) \leq 0 \quad \text{para todo arco } ij.$$

Exemplo 2: 0-potencial mais interessante



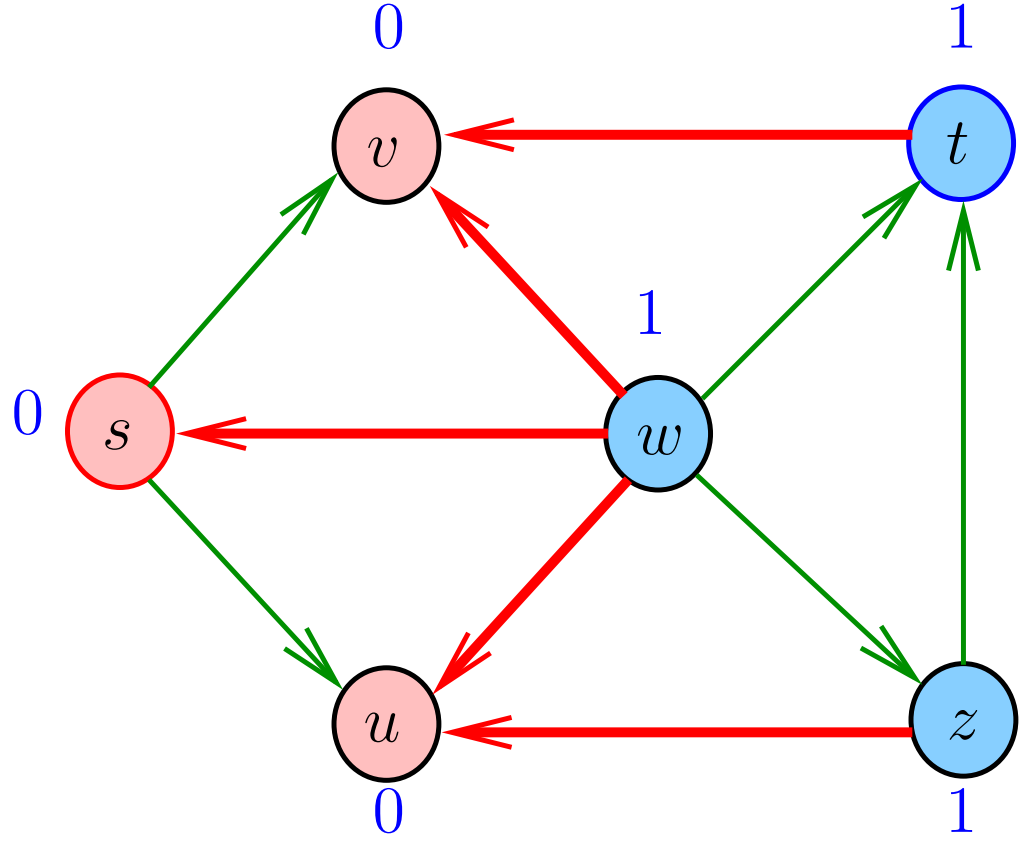
0-Potenciais e cortes

Qualquer 0-potencial y define um corte vazio: se

$$T := \{j \in N : y(j) = 1\}$$

então $\nabla(N - T, T)$ é esse corte.

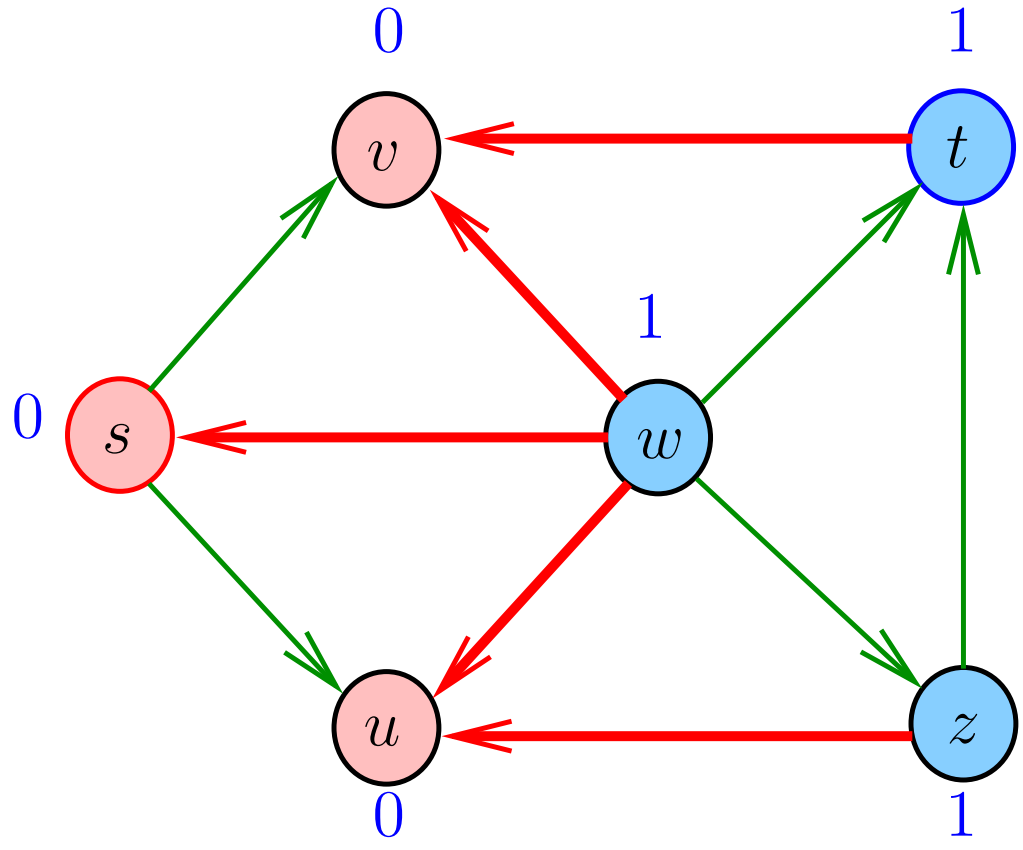
Exemplo 2: corte determinado por um 0-potencial



Propriedade de 0-Potenciais

Se y é um 0-potencial e existe um passeio de s a t então

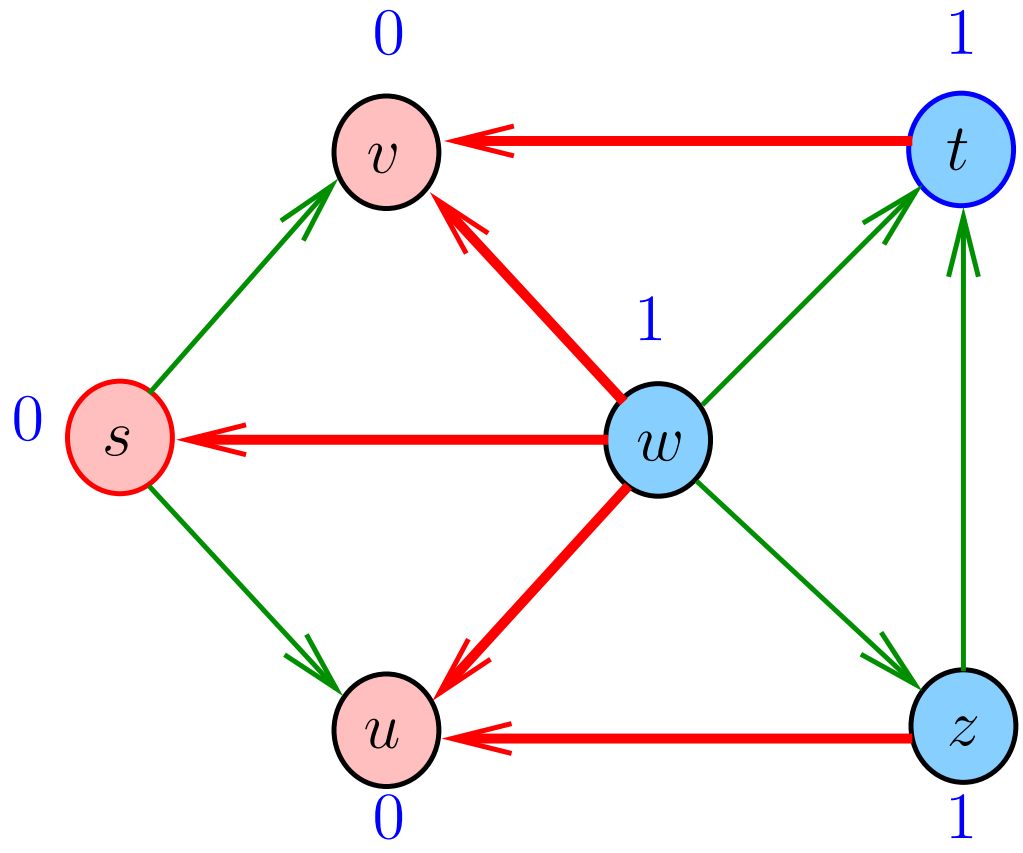
$$y(t) - y(s) \leq 0.$$



Propriedade de 0-Potenciais

Para mostrar que não existe um caminho de s a t basta exibir um 0-potencial tal que

$$y(t) - y(s) > 0.$$



Busca genérico (2)

Recebe dois nós s e t de um grafo (N, A) e devolve um caminho de s a t ou um 0-potencial y tal que $y(t) - y(s) > 0$.

BUSCA-GENÉRICO (N, A, s, t)

0 **para cada** i em N **faça**
1 $\quad y(i) \leftarrow 1$
2 $\quad \pi(i) \leftarrow \text{NIL}$
3 $y(s) \leftarrow 0$
4 **enquanto** $y(j) > y(i)$ **para** algum $ij \in A$ **faça**
5 $\quad y(j) \leftarrow y(i)$
6 $\quad \pi(j) \leftarrow i$
7 **se** $y(t) = 0$
8 $\quad$ **então devolva** o st-caminho no grafo (N, A_π)
9 $\quad$ **senão devolva** y

Invariantes

Na linha 4, antes da verificação da condição "$y(j) > y(i) \ldots$" valem as seguintes invariantes:

(i1) para cada arco pq no grafo de predecessores tem-se $y(p) = y(q) = 0$;

(i2) $\pi(s) = \text{NIL}$ e $y(s) = 0$;

(i3) para cada nó v distinto de s, $y(v) = 0 \Leftrightarrow \pi(v) \neq \text{NIL}$;

(i4) para cada nó v, se $\pi(v) \neq \text{NIL}$ então existe um caminho de s a v no grafo de predecessores.

Correção

Início da última iteração:

- y é um 0-potencial
- se $y(t) = 0$ então (por (i3)) $\pi(t) \neq \text{NIL}$, logo (por (i4)) existe caminho de s a t
- se $y(t) = 1$ então (por (i2)) $y(t) - y(s) > 0$

Conclusão: o algoritmo faz o que promete.

Demonstrações

Demonstração de (i1): Nenhum arco no grafo de predecessores tem ponta inicial ou ponta final igual a j.

Demonstração de (i2): Basta observar que $j \neq s$.

Demonstração de (i3): Os únicos valores de y e π que são alterados por uma iteração são os de j.

Mais demonstração

Demonstração de (i4): Suponha $\pi(v) \neq \text{NIL}$ no início da iteração.

Existe caminho P de s a v no grafo de predecessores (por (i4)).

Temos $y(k) = 0$ para cada nó k de P (por (i1)).

No fim da iteração, P continua sendo um caminho de s a v no grafo de predecessores.

Quando $v = i$, $P \cdot (i, j)$ é uma caminho no grafo de predecessores.

Conclusão

Para quaisquer nós s e t de um grafo (N, A), vale uma e apenas umas das seguintes afirmações:

- existe um caminho de s a t
- existe um 0-potencial y tal que $y(t) - y(s) > 0$.

Consumo de tempo

O número de execuções do bloco de linhas 4–6 é

$$\leq n - 1.$$

linha	consumo de todas as execuções da linha
0-2	$\Theta(n)$
3	$\Theta(1)$
4	$n\mathrm{O}(m)$
5-6	$n\mathrm{O}(1)$
7-9	$\mathrm{O}(n)$
total	$\Theta(n+1) + n\mathrm{O}(m+n+1)$ $= \mathrm{O}(nm)$

Conclusão

O consumo de tempo do algoritmo BUSCA-GENÉRICO é $O(nm)$.

Implementação do algoritmo genérico

BUSCA (N, A, s, t)

0 **para cada** i em N **faça**
1 $A'(i) \leftarrow A(i)$ $y(i) \leftarrow 1$ $\pi(i) \leftarrow$ NIL
2 $y(s) \leftarrow 0$ $L \leftarrow \{s\}$

3 **enquanto** $L \neq \emptyset$ **faça**
4 escolha um nó i em L
5 **se** $A'(i) \neq \emptyset$ **então**
6 retire um arco ij de $A'(i)$
7 **se** $y(j) = 1$ **então**
8 $y(j) \leftarrow 0$ $\pi(j) \leftarrow i$ $L \leftarrow L \cup \{j\}$
9 **senão** $L \leftarrow L - \{i\}$

10 **se** $y(t) = 0$
11 **então devolva** o st-caminho no grafo (N, A_π)
12 **senão devolva** y

Invariantes

Na linha 3, antes da verificação da condição "$L \neq \emptyset$" valem, além de (i1)–(i4) as seguintes invariantes:

(i5) para cada arco pq, se $y(p) = 0$ e $y(q) = 1$ então $p \in L$;

(i6) $y(p) = 0$ para cada p em L;

(i7) para cada nó p e cada arco pq em $A(p) - A'(p)$, se $y(p) = 0$ então $y(q) = 0$.

Correção

No início da última iteração:

- Por (i5), $y(q) - y(p) \leq 0$ para todo pq; portanto, y é um 0-potencial.
- Se $y(t) = 1$, então (por (i2)) $y(t) - y(s) = 1 > 0$.
- Senão (por (i4)), há caminho de s a t.

Conclusão: o algoritmo faz o que promete.

Problema da Busca (continuação)

PF 3.3, 3.4

Consumo de tempo

Note que:

- $\leq n$ iterações passam pela linha 9
- $\leq \sum_{k \in N} |A(k)| = m$ iterações passam pelas linhas 6, 7, e 8
- consumo de tempo de cada iteração é $\mathrm{O}(1)$

Conclusão: o consumo de total de tempo das iterações das linhas 3–9 é $(n+m)\mathrm{O}(1) = \mathrm{O}(n+m)$.

Consumo de tempo

O número de execuções do bloco de linhas 3–6 é

$$\leq n + m.$$

linha	consumo de todas as execuções da linha
0–1	$\mathrm{O}(n)$
2	$\mathrm{O}(1)$
3–9	$(n+m)\mathrm{O}(1) = \mathrm{O}(n+m)$
10–12	$\mathrm{O}(n)$
total	$\mathrm{O}(n+1) + \mathrm{O}(2n+m)$ $= \mathrm{O}(n+m)$

Conclusão

O consumo de tempo do algoritmo BUSCA é $O(n+m)$.

Busca em largura e em profundidade

- Se L no algoritmo BUSCA for manipulada como uma fila teremos um

  algoritmo de busca em largura

  ($=$ *breadth-first search* $=$ *BFS*).

- Se L no algoritmo BUSCA for manipulada como uma pilha teremos um

  algoritmo de busca em profundidade

  ($=$ *depth-first search* $=$ *DFS*).