Melhores momentos

AULA 16

Programação dinâmica

Propriedade (da subestrutura ótima)
Se G é um digrafo com custos não-negativos nos arcos e v_0-v_1-v_2-...-v_k é um caminho mínimo então v_i-v_{i+1}-...-v_j é um caminho mínimo para $0 \leq i \leq j \leq k$

custo[v][w] = menor custo de uma caminho de v a w

Propriedade 1
O valor de custo[s][t] é

$$\min\{\texttt{custo[s][v]} + \texttt{custo[v][t]} : \texttt{v} \text{ é vértice}\}$$

Programação dinâmica

Propriedade (da subestrutura ótima)
Se G é um digrafo com custos não-negativos nos arcos e v_0-v_1-v_2-...-v_k é um caminho mínimo então v_i-v_{i+1}-...-v_j é um caminho mínimo para $0 \leq i \leq j \leq k$

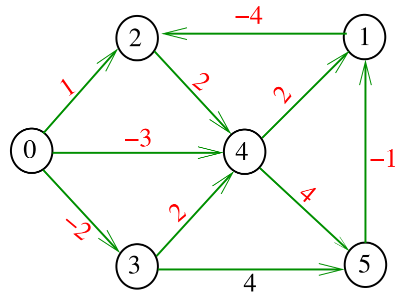
custo[v][w] = menor custo de uma caminho de v a w

Propriedade 2
O valor de custo[s][t] é

$$\min\{\texttt{custo[s][v]} + \texttt{G->adj[v][t]} : \texttt{v-t} \text{ é arco}\}$$

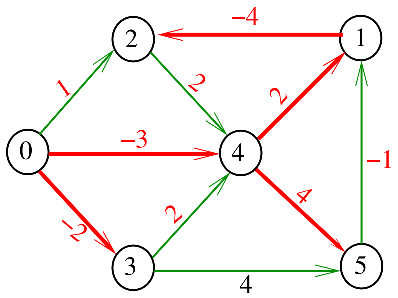
Problema da SPT

Problema: Dado um vértice s de um digrafo com custos (possivelmente negativos) nos arcos, encontrar uma SPT com raiz s
Entra:

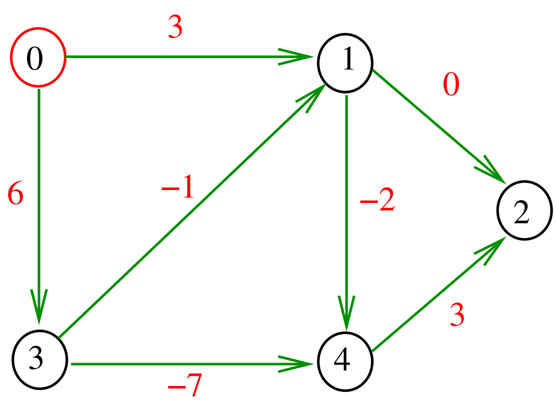


Problema da SPT

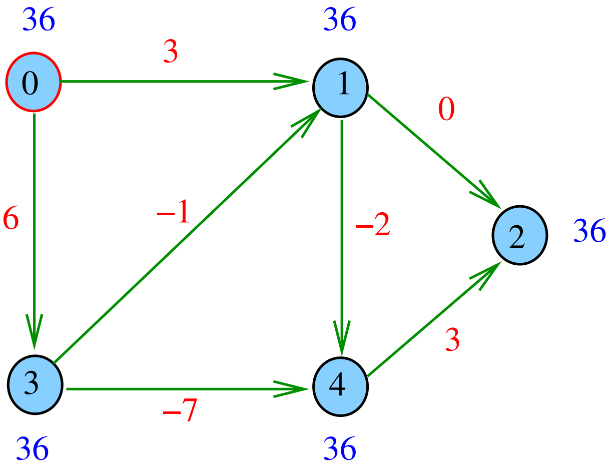
Problema: Dado um vértice s de um digrafo com custos (possivelmente negativos) nos arcos, encontrar uma SPT com raiz s
Sai:



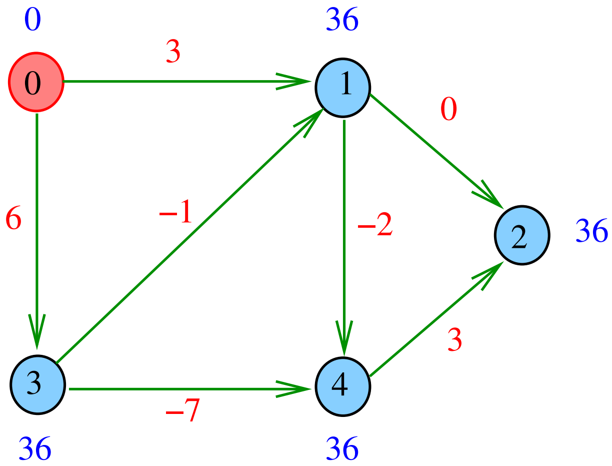
Apelemos para Dijkstra



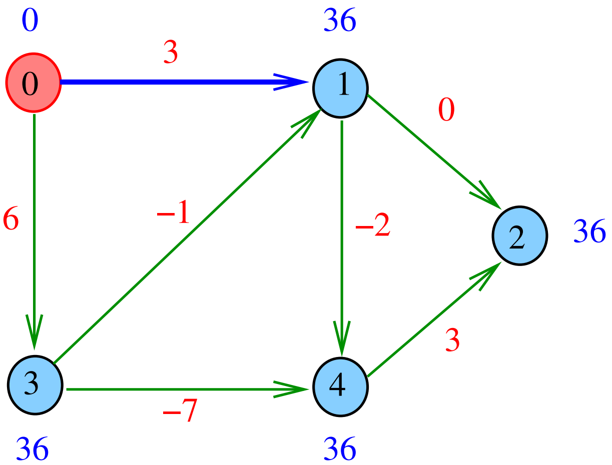
Apelemos para Dijkstra



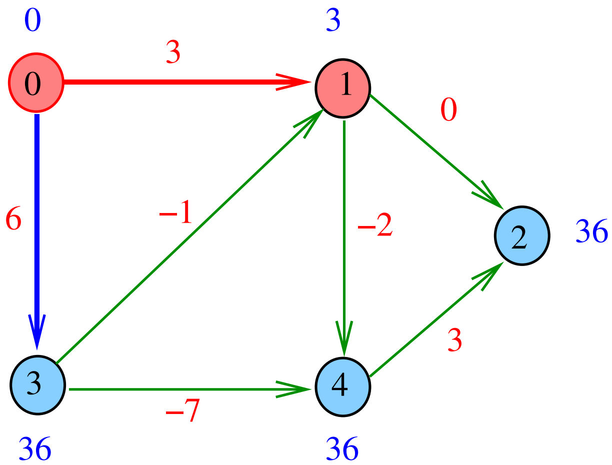
Apelemos para Dijkstra



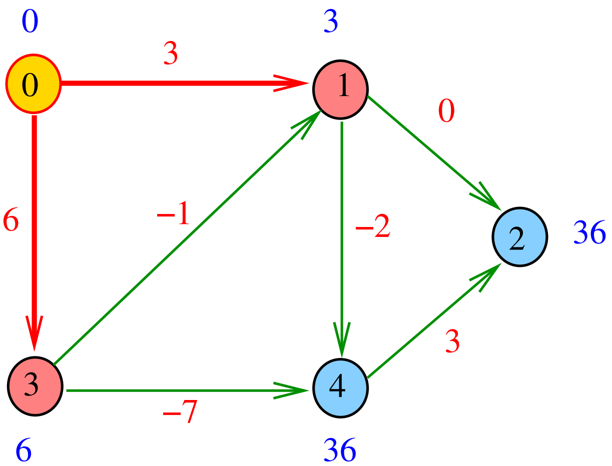
Apelemos para Dijkstra



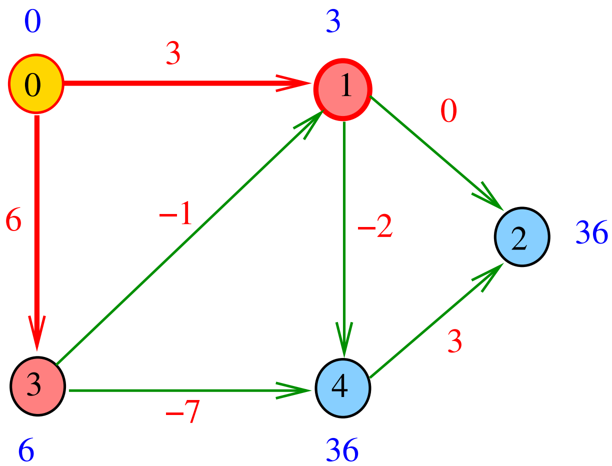
Apelemos para Dijkstra



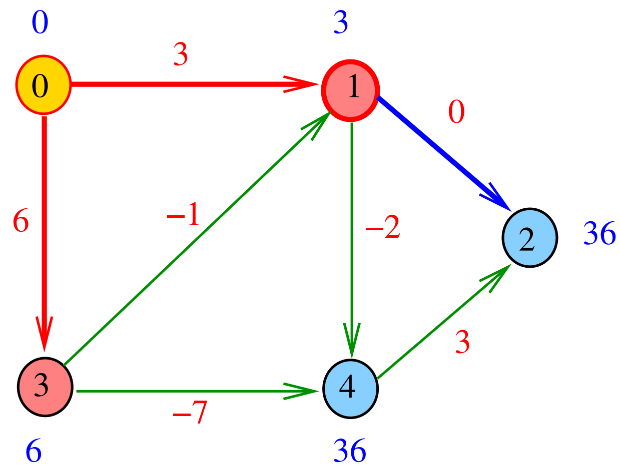
Apelemos para Dijkstra



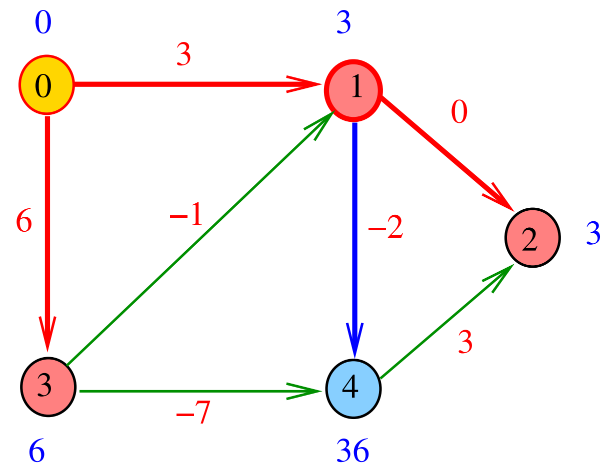
Apelemos para Dijkstra



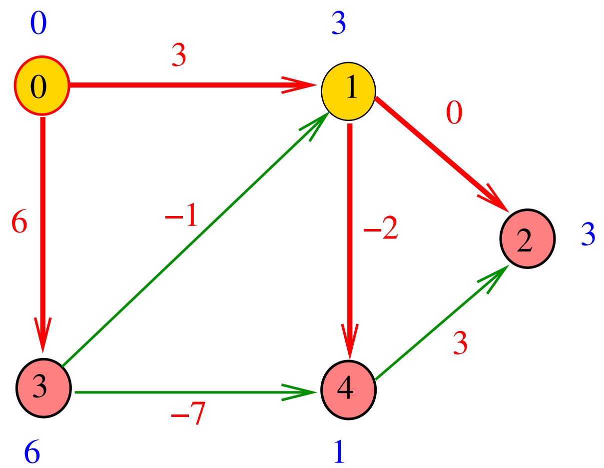
Apelemos para Dijkstra



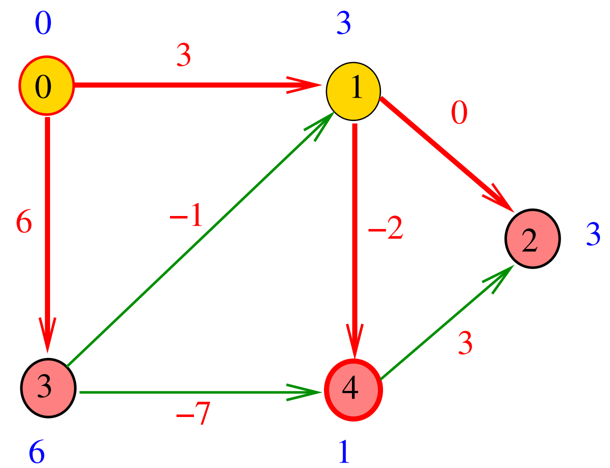
Apelemos para Dijkstra



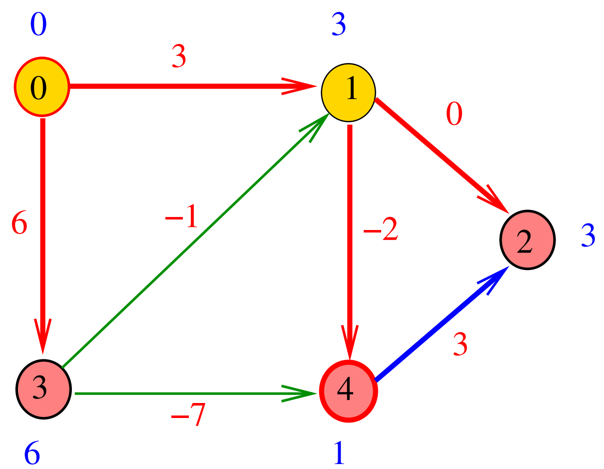
Apelemos para Dijkstra



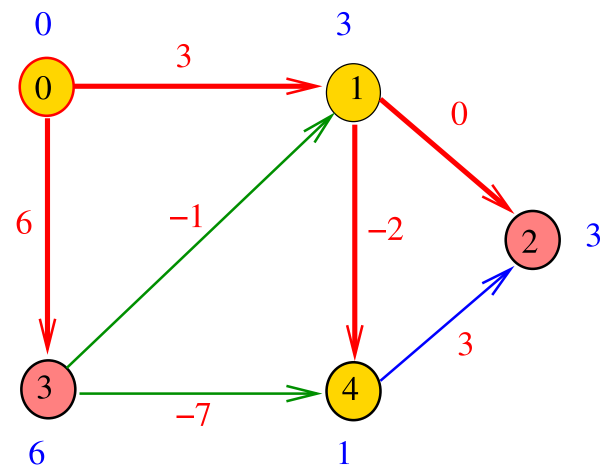
Apelemos para Dijkstra



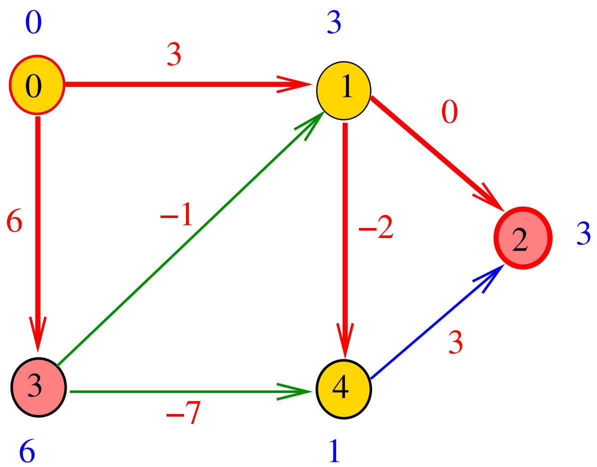
Apelemos para Dijkstra



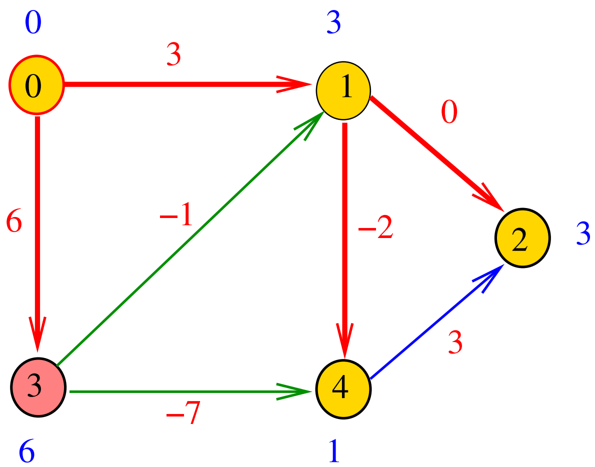
Apelemos para Dijkstra



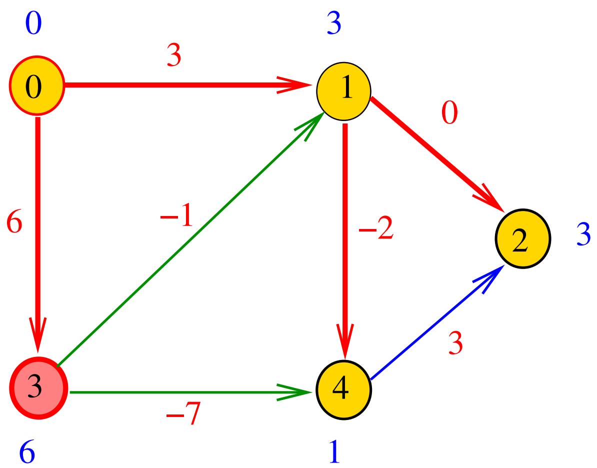
Apelemos para Dijkstra



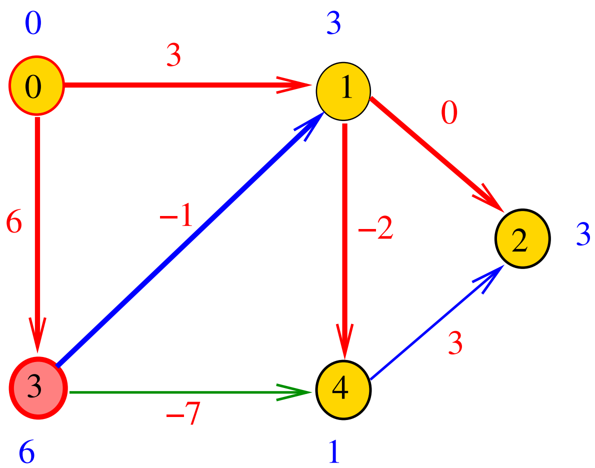
Apelemos para Dijkstra



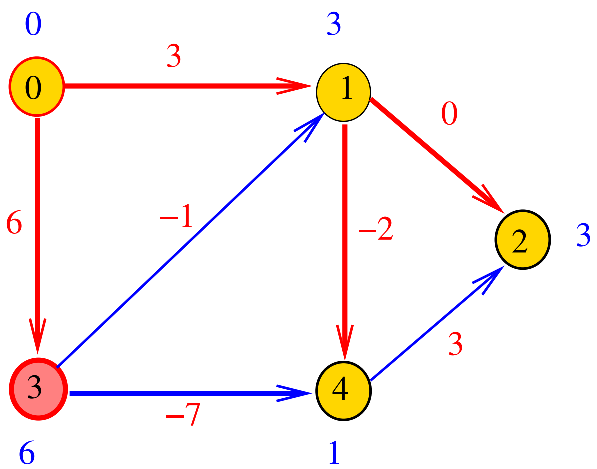
Apelemos para Dijkstra



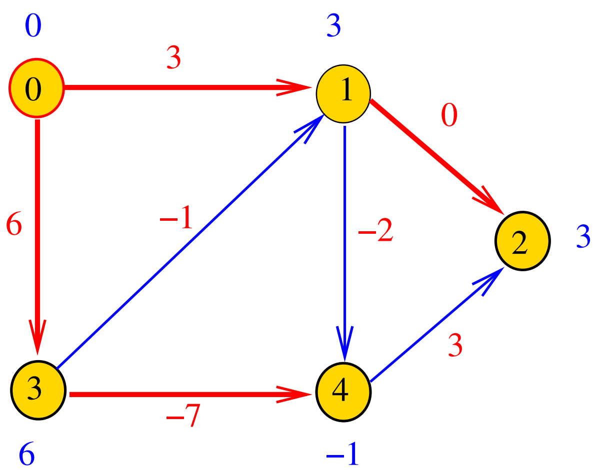
Apelemos para Dijkstra



Apelemos para Dijkstra



Opsss



O caminho mínimo de 0 a 2 tem custo 2 e não 3. . .

Conclusão

O algoritmo de Dijkstra não funciona para digrafos com custos negativos, mesmo que o digrafo seja acíclico.

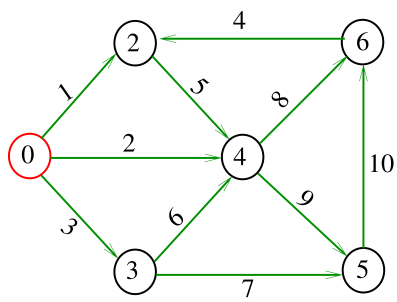
AULA 17

Algoritmo de Bellman-Ford

Problema da SPT

Problema: Dado um vértice s de um digrafo com custos (possivelmente negativos) nos arcos, encontrar uma SPT com raiz s

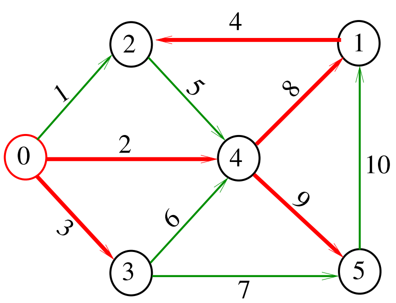
Entra:



Problema da SPT

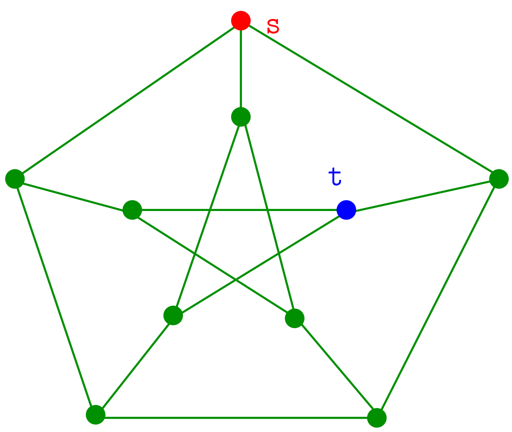
Problema: Dado um vértice s de um digrafo com custos (possivelmente negativos) nos arcos, encontrar uma SPT com raiz s

Sai:



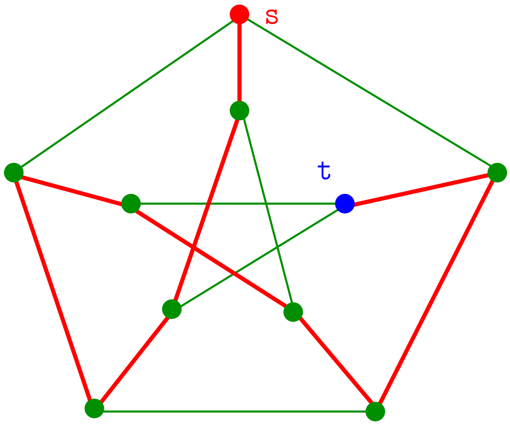
Caminhos hamiltonianos

Problema: Dados vértices s e t de um grafo encontrar um caminho hamiltoniano de s e t



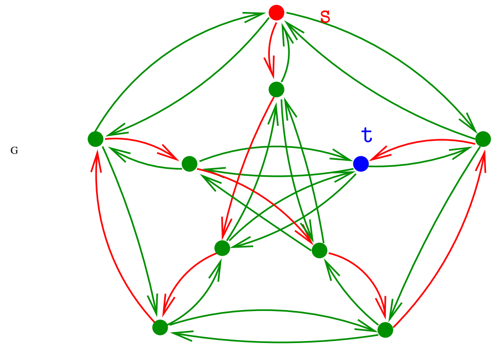
Caminhos hamiltonianos

Problema: Dados vértices s e t de um grafo encontrar um caminho hamiltoniano de s e t



Redução polinomial

todos custos $= -1$



G possui um st-caminho hamiltoniano $\Leftrightarrow$
G possui um st-caminho **simples** de custo $-(V - 1)$.

Conclusão

É sabido que:

O problema do caminho hamiltoniano é NP-difícil.

Assim, como problema do caminho simples mínimo com custos negativos é tão difícil quanto o problema do caminho hamiltoniano.

O problema do caminho simples de custo mínimo é NP-difícil.

Complexidade computacional

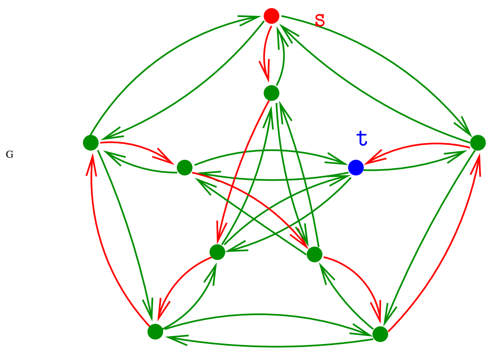
O problema do caminho simples de custo mínimo é NP-difícil.

NP-difícil = **não se conhece** algoritmo de consumo de 'tempo polinomial'

Em outras palavras: ninguém conhece um algoritmo eficiente para o problema ...
Se alguém conhece, não contou para ninguém ...

Subestrutura ótima ...

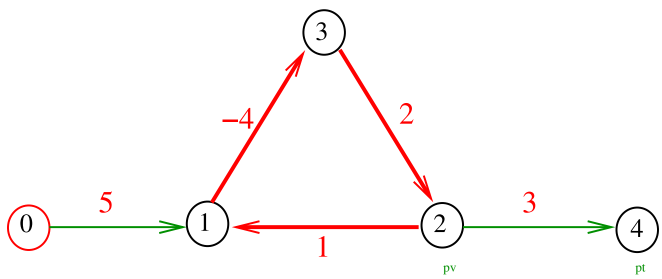
todos custos $= -1$



Não vale para digrafos com ciclos negativos

Ciclos negativos

Se o digrafo possui um **ciclo (de custo) negativo** alcançável a partir de s, então não existe caminho mínimo de s a alguns vértices



Vamos supor que o digrafo não tem ciclos negativos.

Caminho de custo mínimo

Problema: Dado vértices s e t de um digrafo com custos (**possivelmente negativos**) nos arcos, encontrar uma caminho de custo mínimo de s a t.

Esse problema pode ser:

- inviável: não existe caminho de s a t;
- viável e limitado: existe caminho de s a t e nenhum caminho de s a t passa por um ciclo negativo; ou
- viável e ilimitado: existe caminho de s a t mas não existe um caminho de s a t de custo mínimo, ou seja, existe caminho de s a t que passa por um ciclo negativo.

Caminho simples de custo mínimo

Problema: Dado vértices s e t de um digrafo com custos (**possivelmente negativos**) nos arcos, encontrar uma caminho **simples** de custo mínimo de s a t.

Esse problema pode ser:

- inviável: não existe caminho de s a t; ou
- viável (e limitado): existe caminho de s a t.

Este problema é NP-difícil.
Só sabemos resolver esse problema quando o caminho de custo mínimo é **simples**. . .

Programação dinâmica

custo[k][w] = menor custo de um caminho de s a w com $\leq$ k arcos.

Recorrência

Programação dinâmica

custo[k][w] = menor custo de um caminho de s a w com $\leq$ k arcos.

Recorrência

$$
\begin{aligned}
\texttt{custo}[0][\texttt{s}] &= 0\\
\texttt{custo}[0][\texttt{w}] &= \texttt{INFINITO},\ \texttt{w} \neq \texttt{s}\\
\texttt{custo}[\texttt{k}][\texttt{w}] &= \min\{\texttt{custo}[\texttt{k}-1][\texttt{w}],\\
&\qquad \min\{\texttt{custo}[\texttt{k}-1][\texttt{v}] + \texttt{G->adj}[\texttt{v}][\texttt{w}]\}\}
\end{aligned}
$$

Programação dinâmica

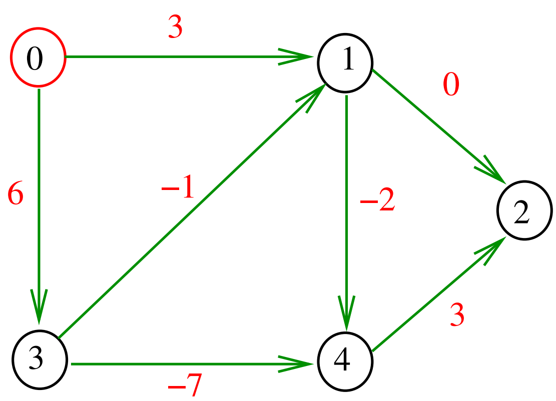
custo[k][w] = menor custo de um caminho de s a w com $\leq$ k arcos.

Recorrência

$$
\begin{aligned}
\texttt{custo}[0][\texttt{s}] &= 0\\
\texttt{custo}[0][\texttt{w}] &= \texttt{INFINITO},\ \texttt{w} \neq \texttt{s}\\
\texttt{custo}[\texttt{k}][\texttt{w}] &= \min\{\texttt{custo}[\texttt{k}-1][\texttt{w}],\\
&\qquad \min\{\texttt{custo}[\texttt{k}-1][\texttt{v}] + \texttt{G->adj}[\texttt{v}][\texttt{w}]\}\}
\end{aligned}
$$

Se o digrafo não tem ciclo negativo acessível a partir de s, então custo[V-1][w] é o menor custo de um caminho de s a w

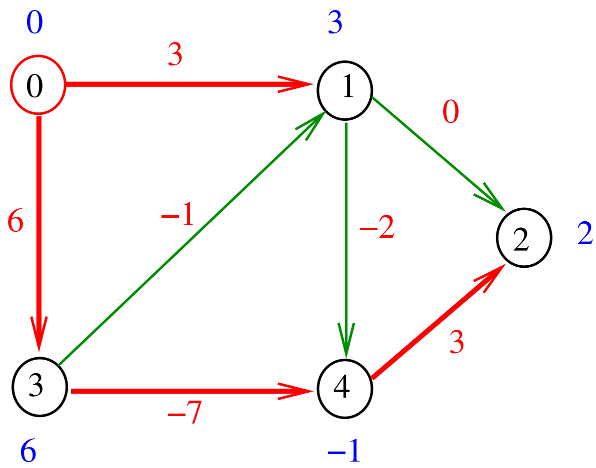
Exemplo



Exemplo

k \ v	0	1	2	3	4
0	0	*	*	*	*
1	0	3	*	6	*
2	0	3	3	6	−1
3	0	3	2	6	−1
4	0	3	2	6	−1

Exemplo




```
void bellman_ford1(Digraph G, Vertex s){
  Vertex v, w; double d;
  for (v=0; v < G->V; v++)
      custo[0][v] = INFINITO;
  custo[0][s] = 0;
  for (k=1; k < G->V; k++)
      for (w=0; w < G->V; w++){
          custo[k][w] = custo[k-1][w];
          for (v=0; v < G->V; v++){
              d=custo[k-1][v]+G->adj[v][w];
              if (custo[k][w] > d)
                  custo[k][w] = d;
          }
      }
}
```

Consumo de tempo

O consumo de tempo da função bellman_ford1 é $O(V^3)$.

Ciclos negativos

Se custo[k][v] $\neq$ custo[k-1][v], então custo[k][v] é o custo de um caminho de s a v com **exatamente** k arcos.

Se custo[V][v] $\neq$ custo[V-1][v], então

- custo[V][v] < custo[V-1][v] e
- custo[V][v] é o custo de um caminho P de s a v com **exatamente** V arcos.

Seja C um ciclo em P e seja P′ o caminho resultante a partir de P após a remoção de C.

Ciclos negativos

Note que P′ tem no $\leq$ V-1 arcos e portanto

$$\begin{aligned} \text{custo}(P') &\geq \text{custo}[V\text{-}1][v] \\ &> \text{custo}[V][v] \\ &= \text{custo}(P) \\ &= \text{custo}(P') + \text{custo}(C). \end{aligned}$$

Logo, C é um ciclo de custo negativo.

Conclusão

Se custo[V][v] $\neq$ custo[V-1][v], então G tem um ciclo negativo alcançável a partir de s.