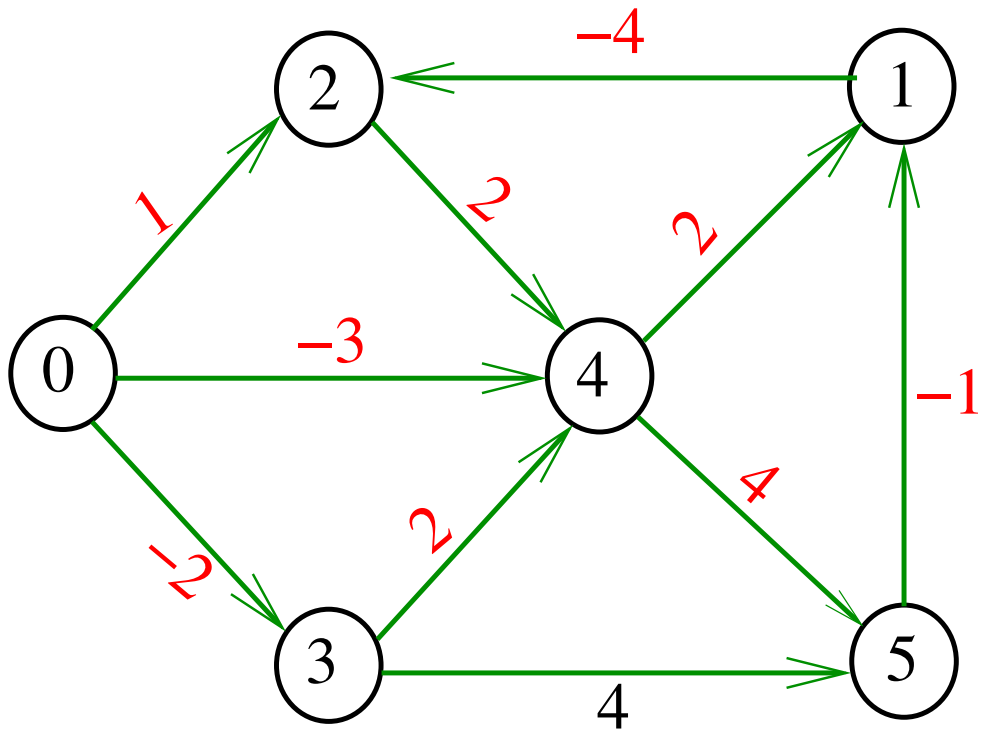
Melhores momentos

AULA 16

Problema da SPT

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

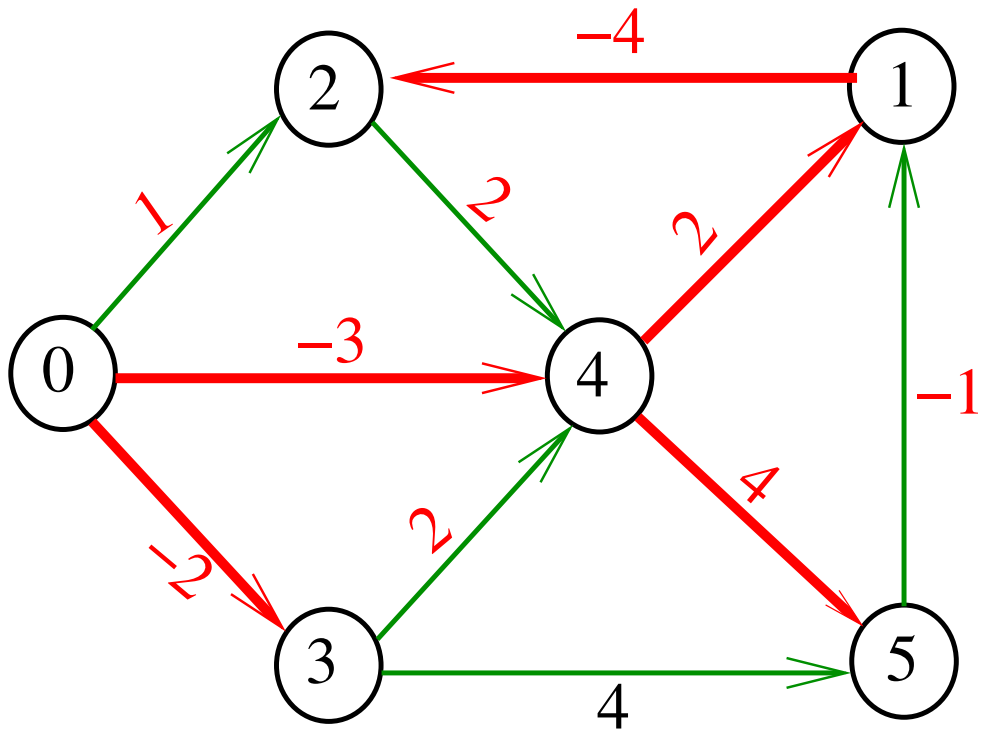
Entra:



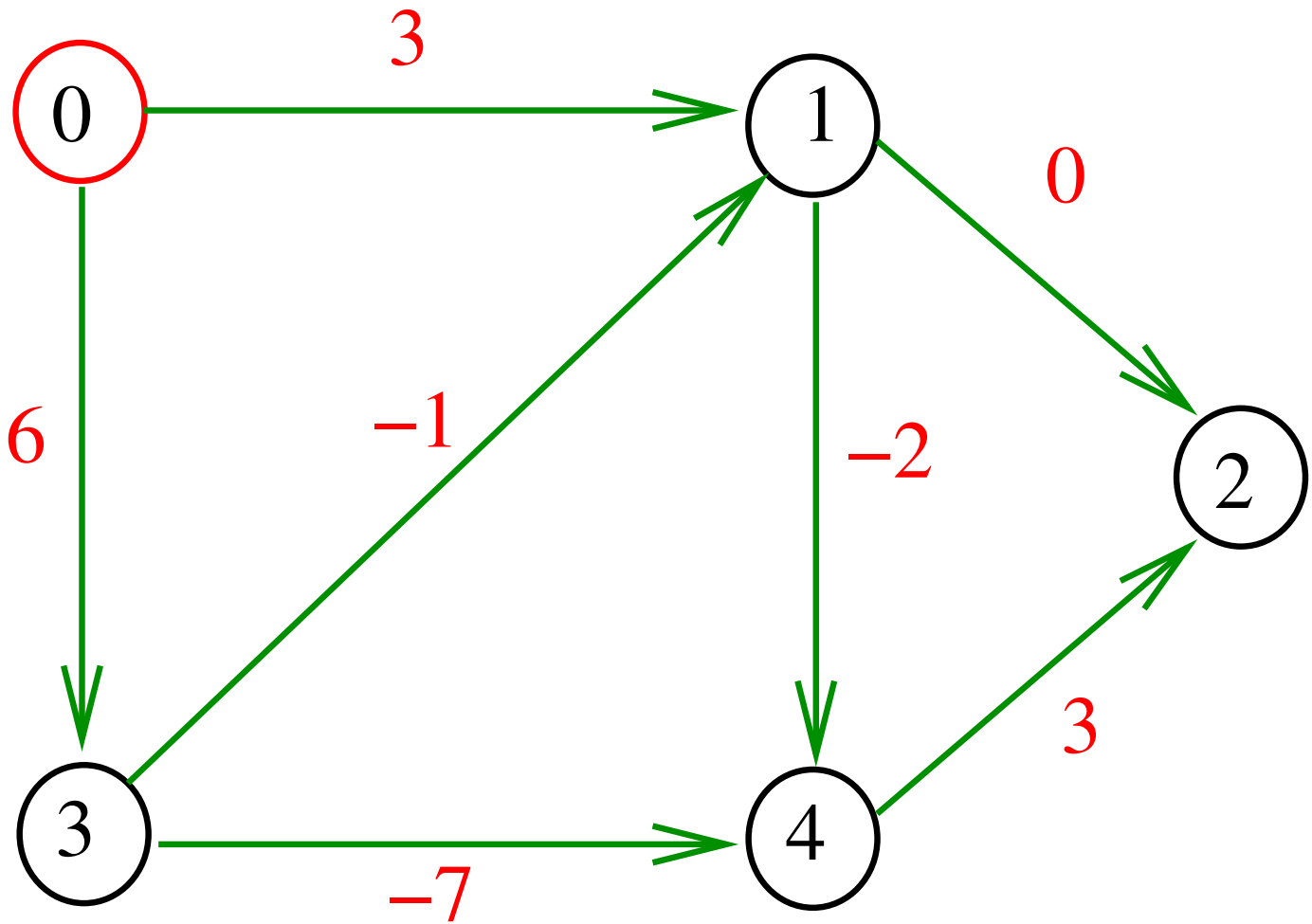
Problema da SPT

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

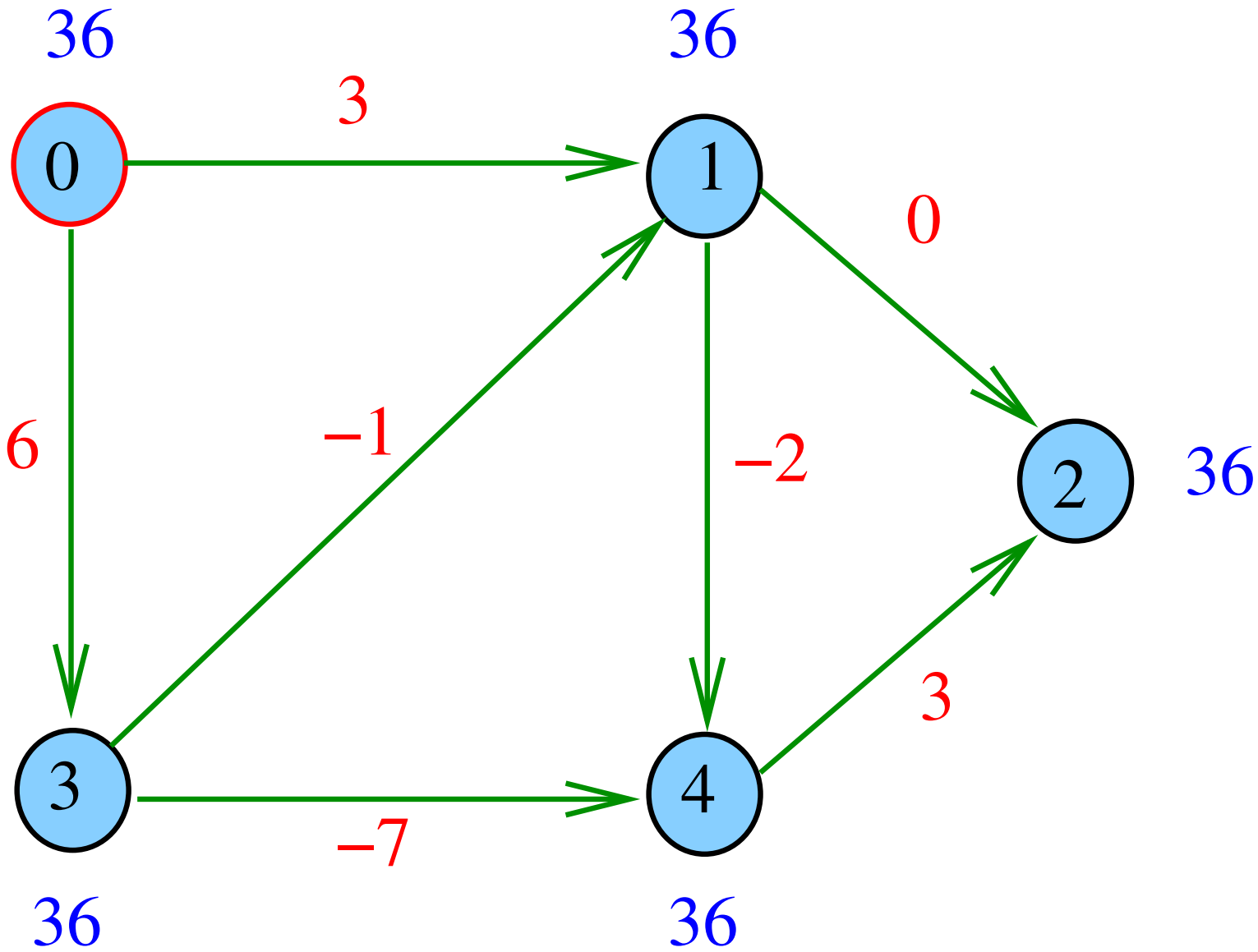
Sai:



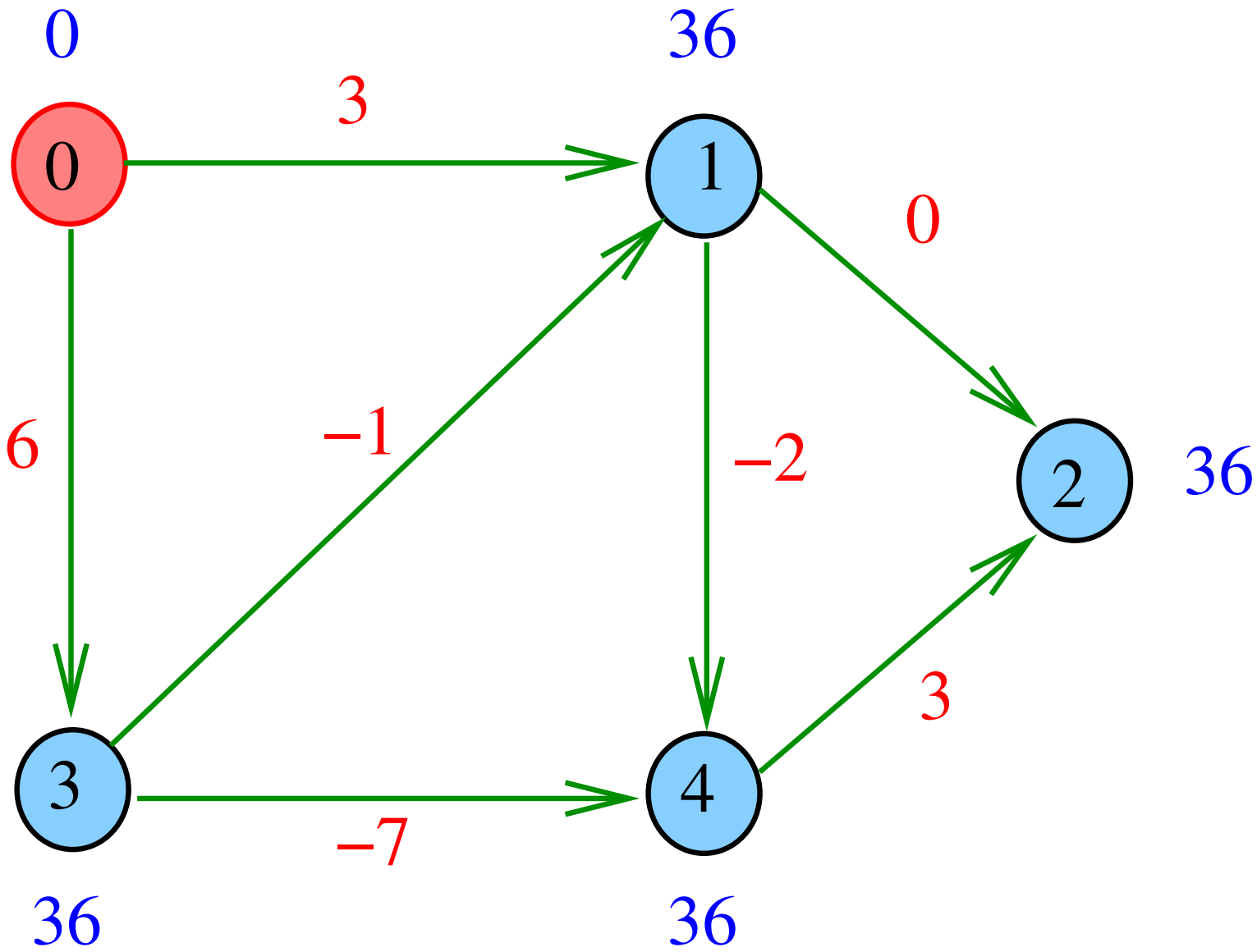
Apelemos para Dijkstra



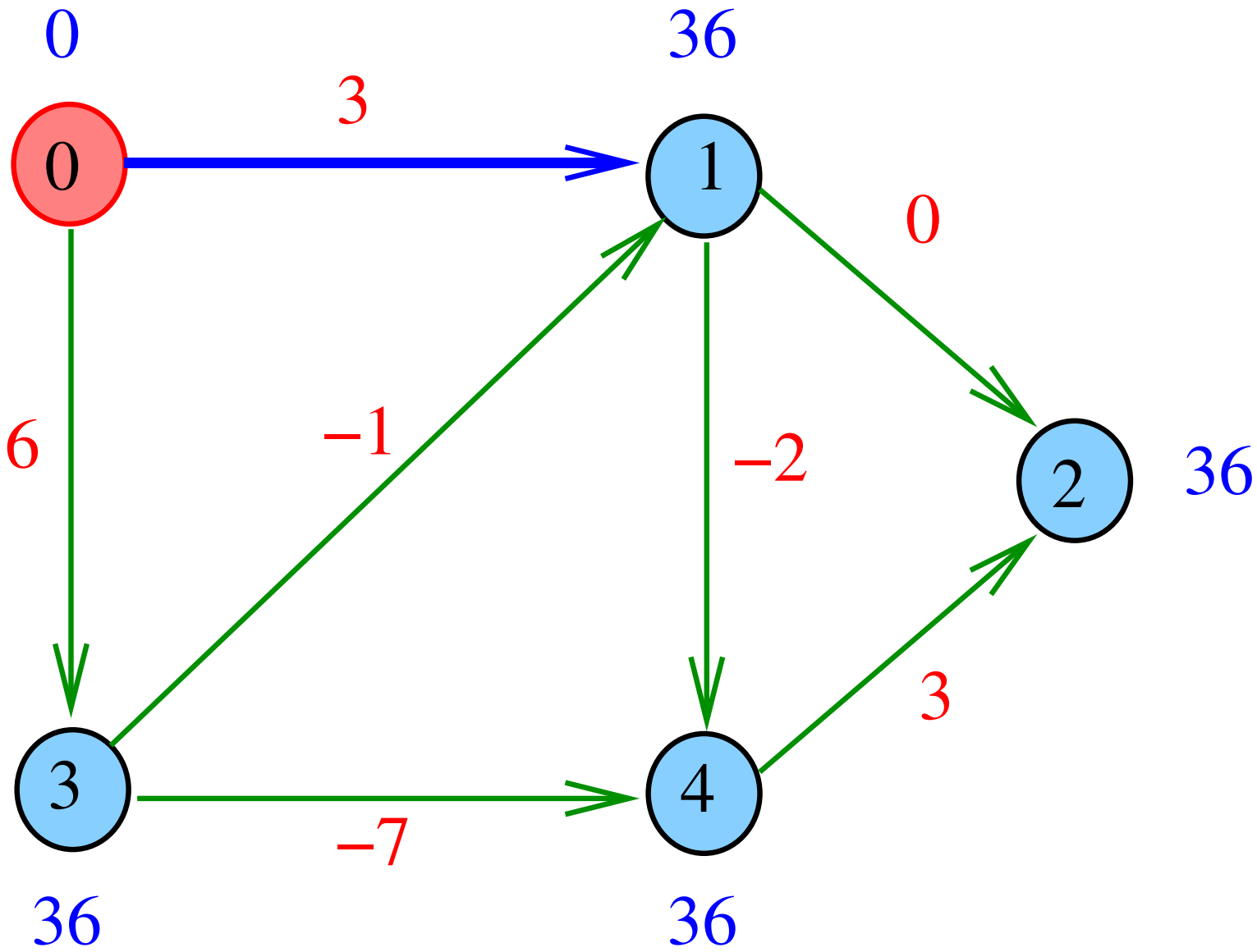
Apelemos para Dijkstra



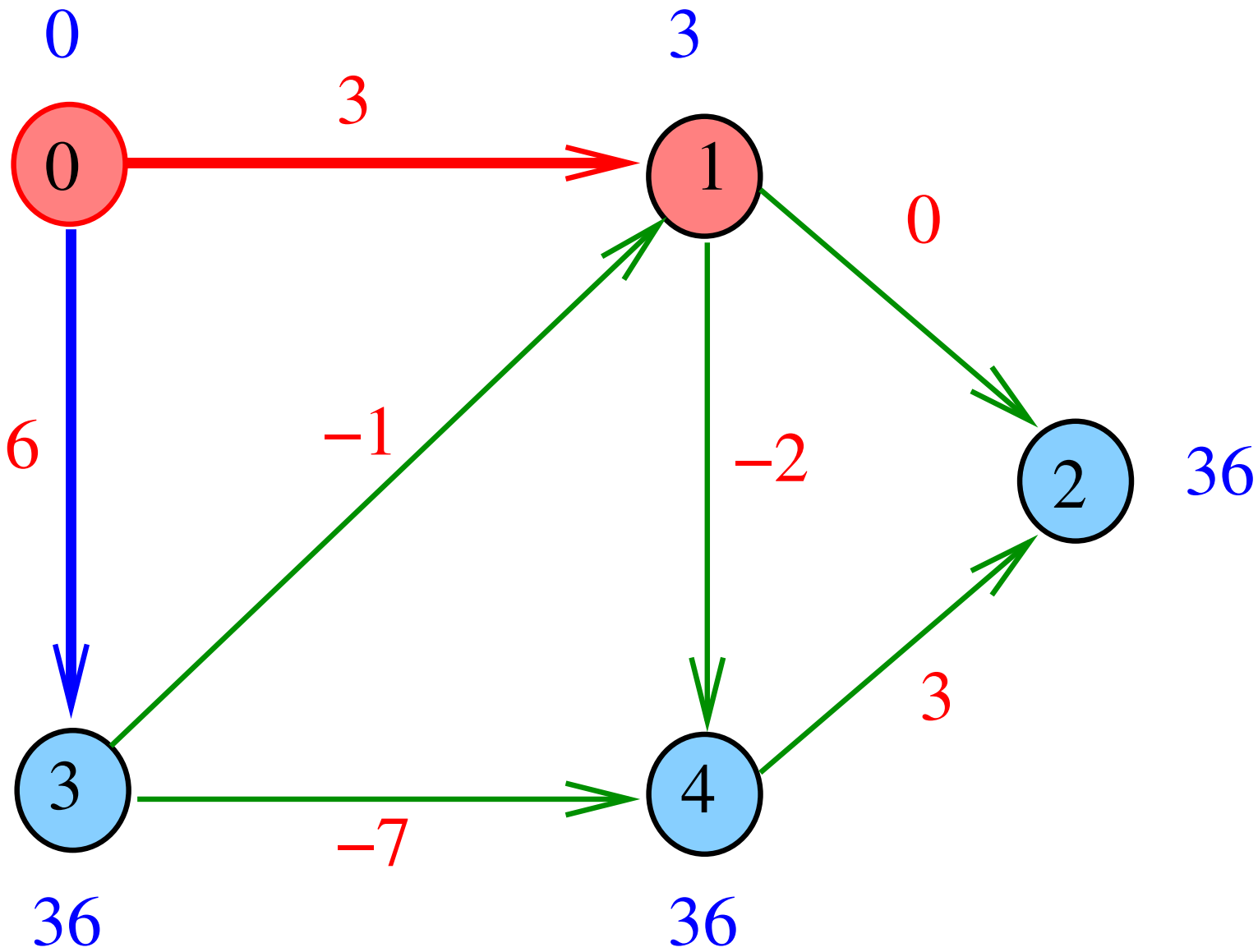
Apelemos para Dijkstra



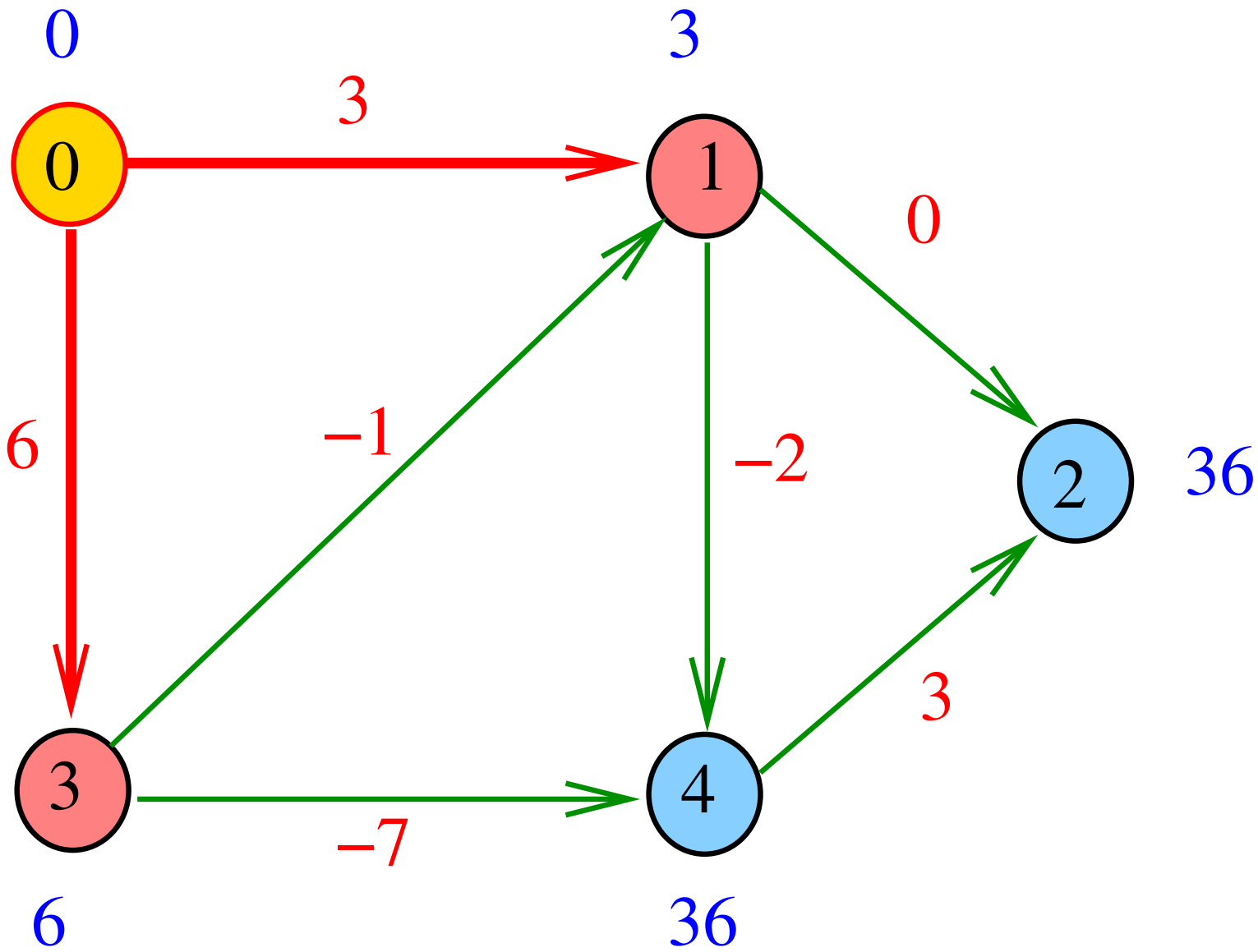
Apelemos para Dijkstra



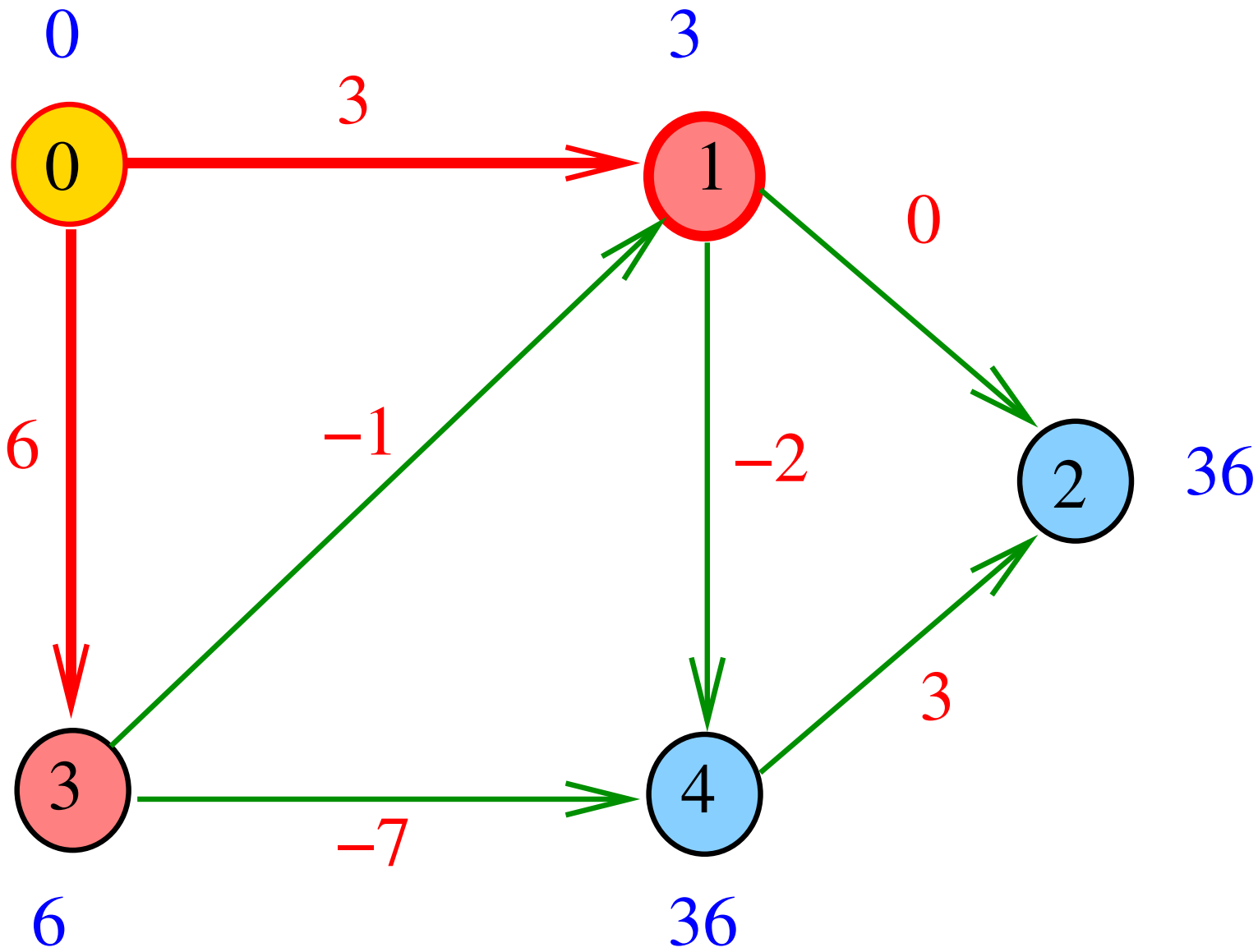
Apelemos para Dijkstra



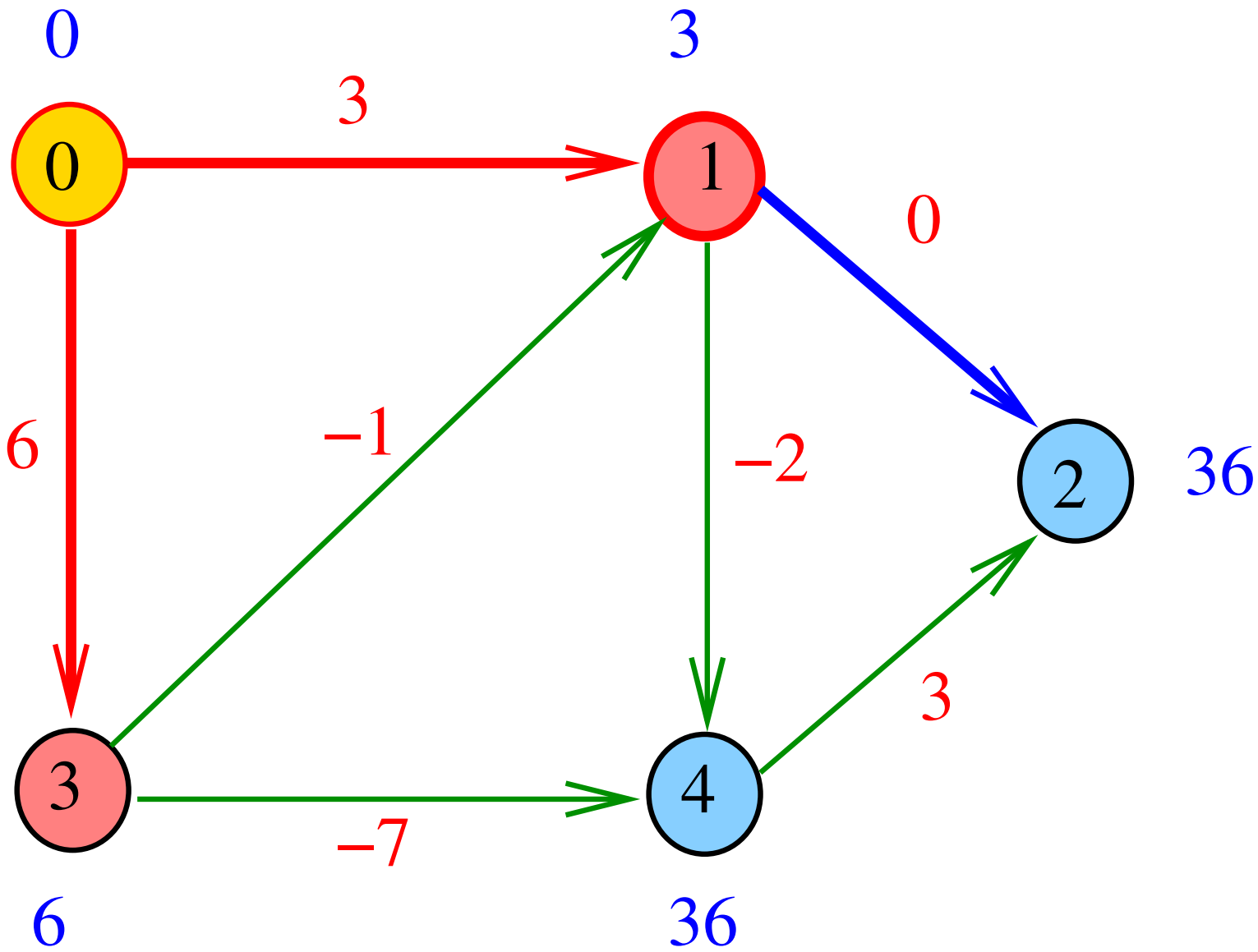
Apelemos para Dijkstra



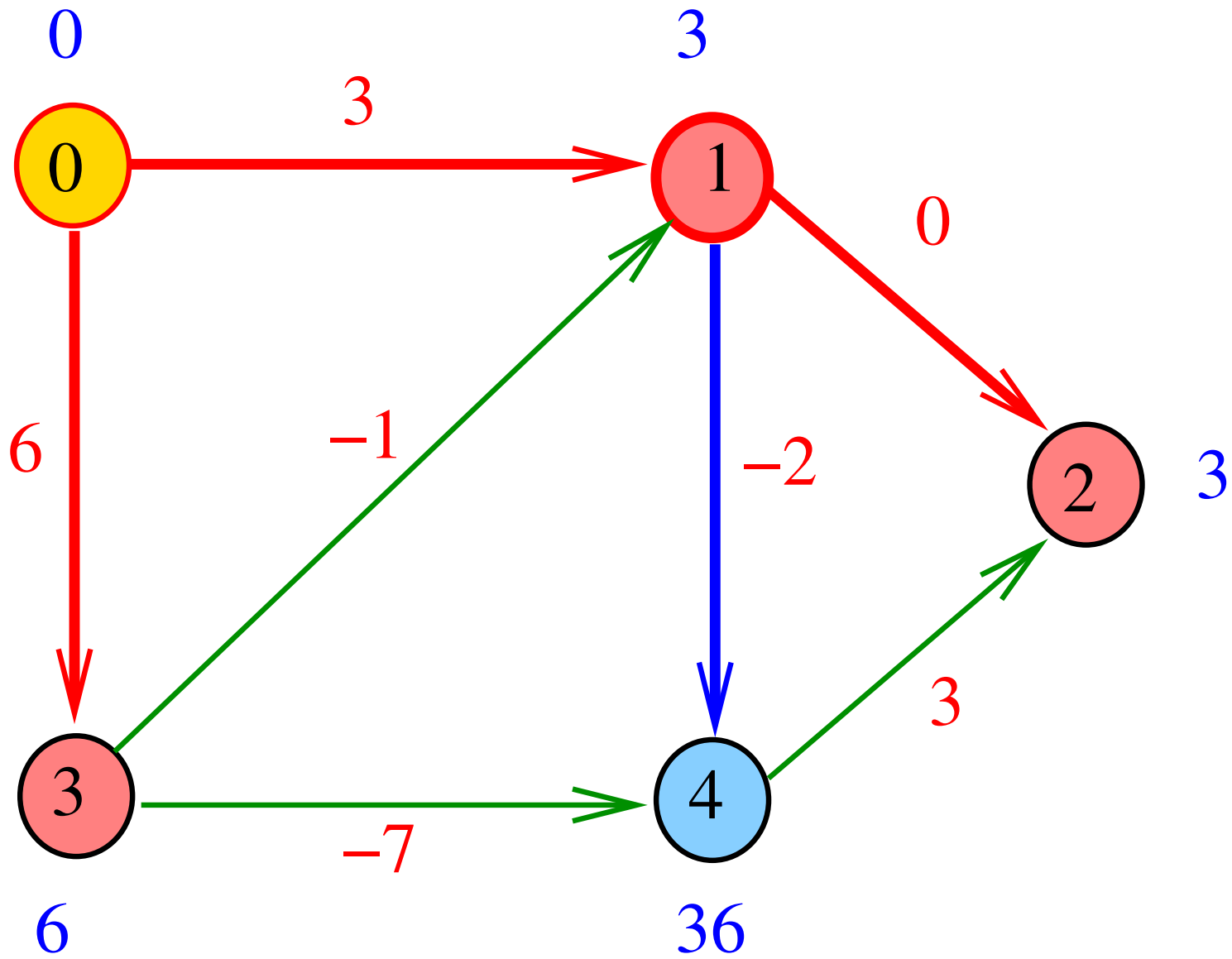
Apelemos para Dijkstra



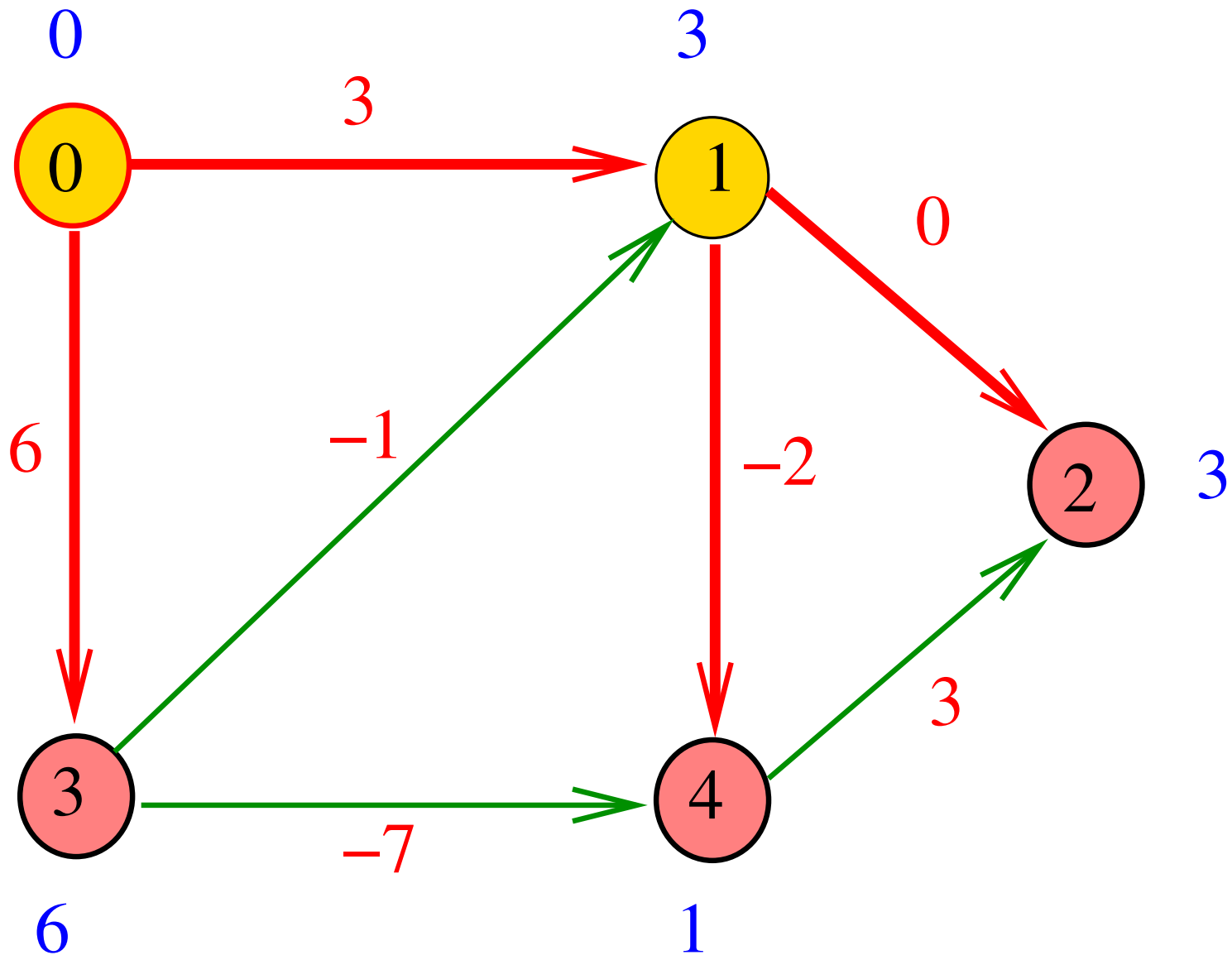
Apelemos para Dijkstra



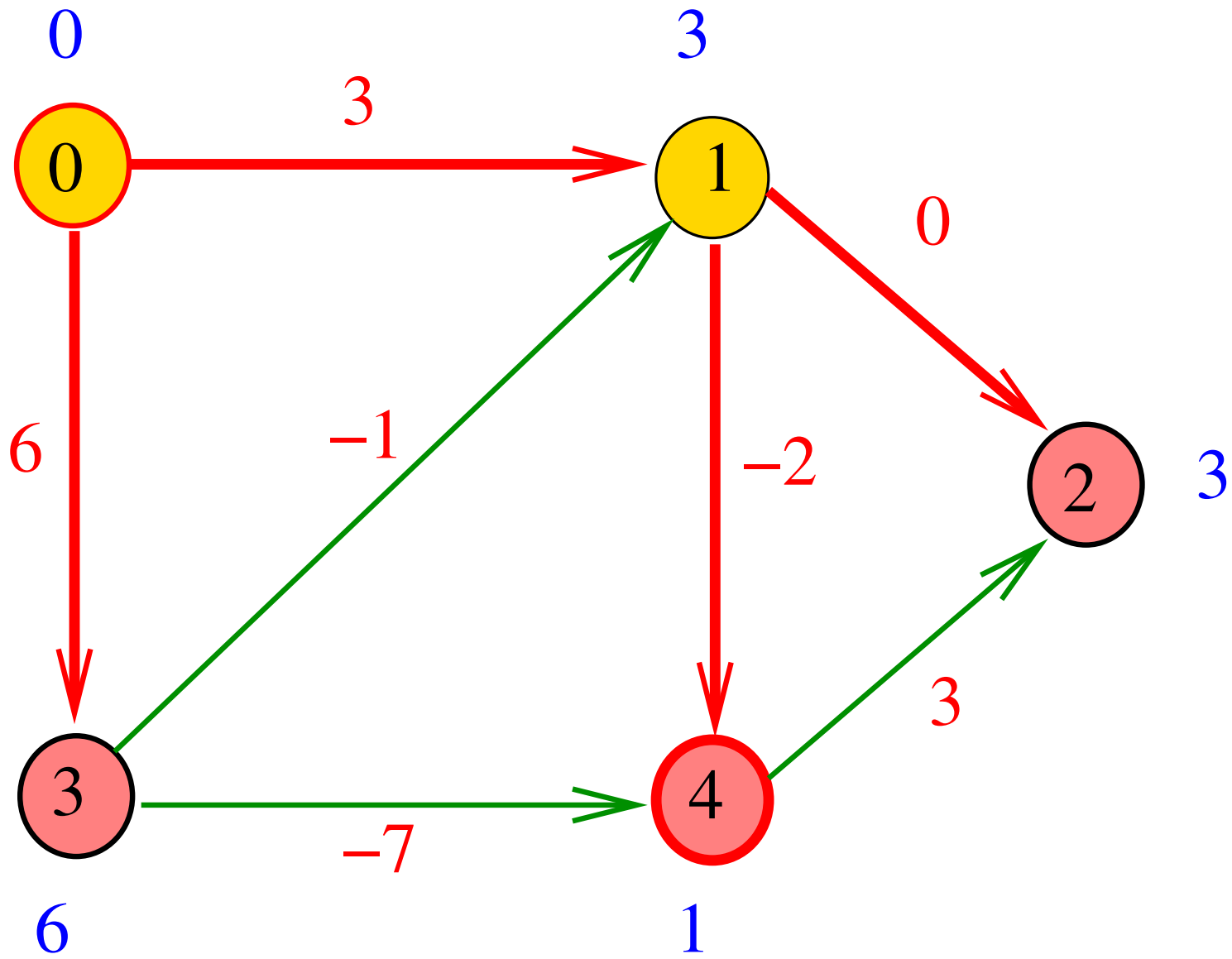
Apelemos para Dijkstra



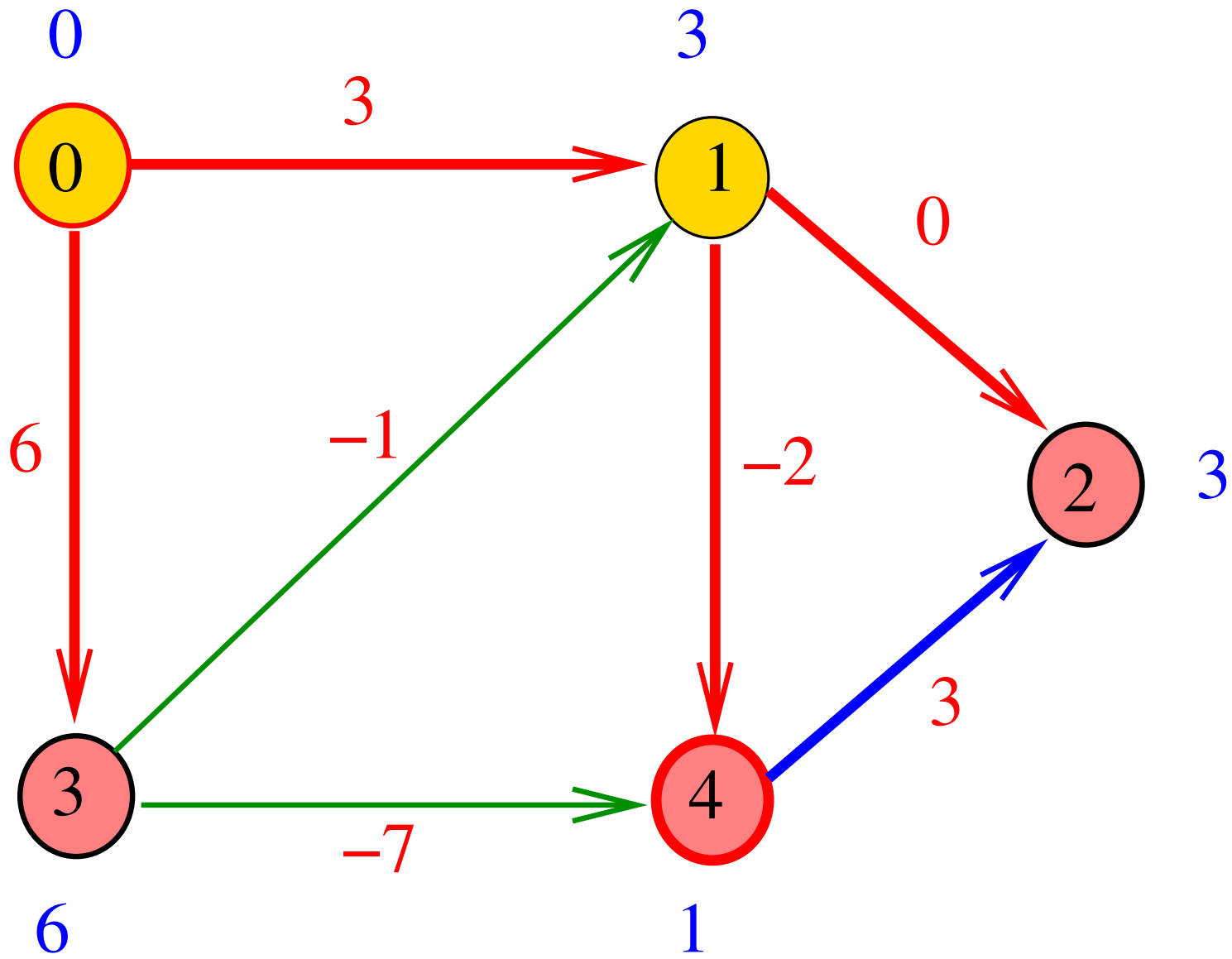
Apelemos para Dijkstra



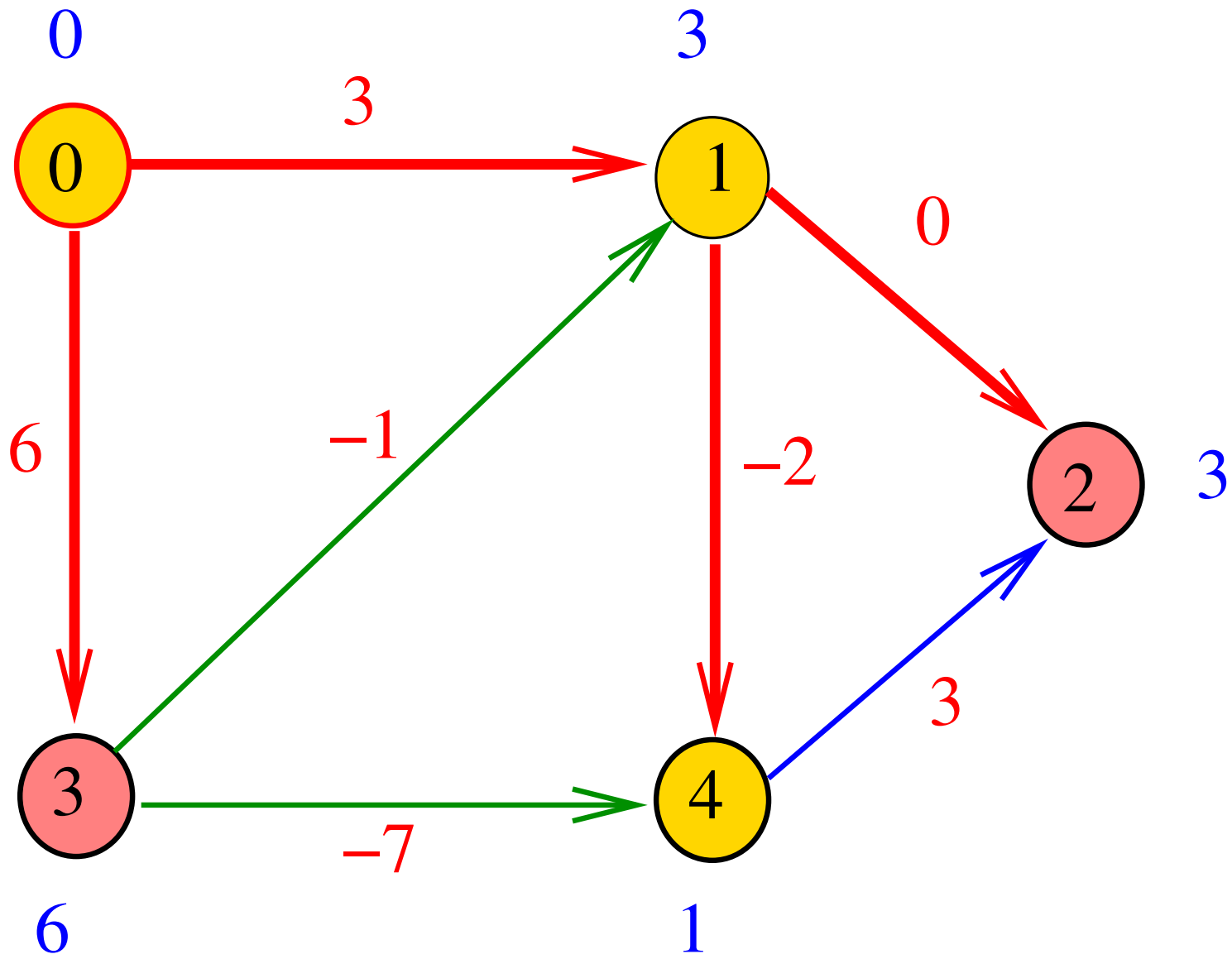
Apelemos para Dijkstra



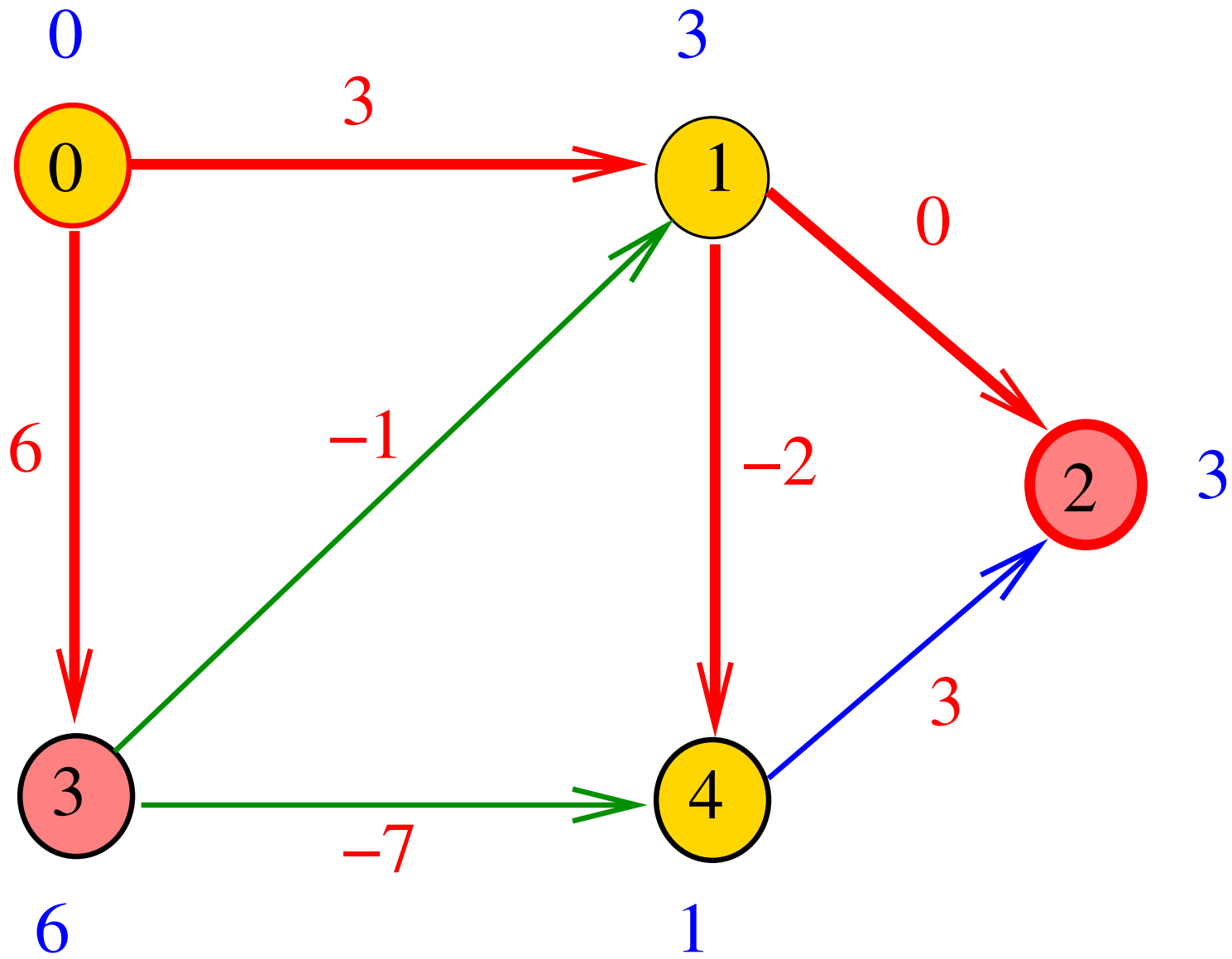
Apelemos para Dijkstra



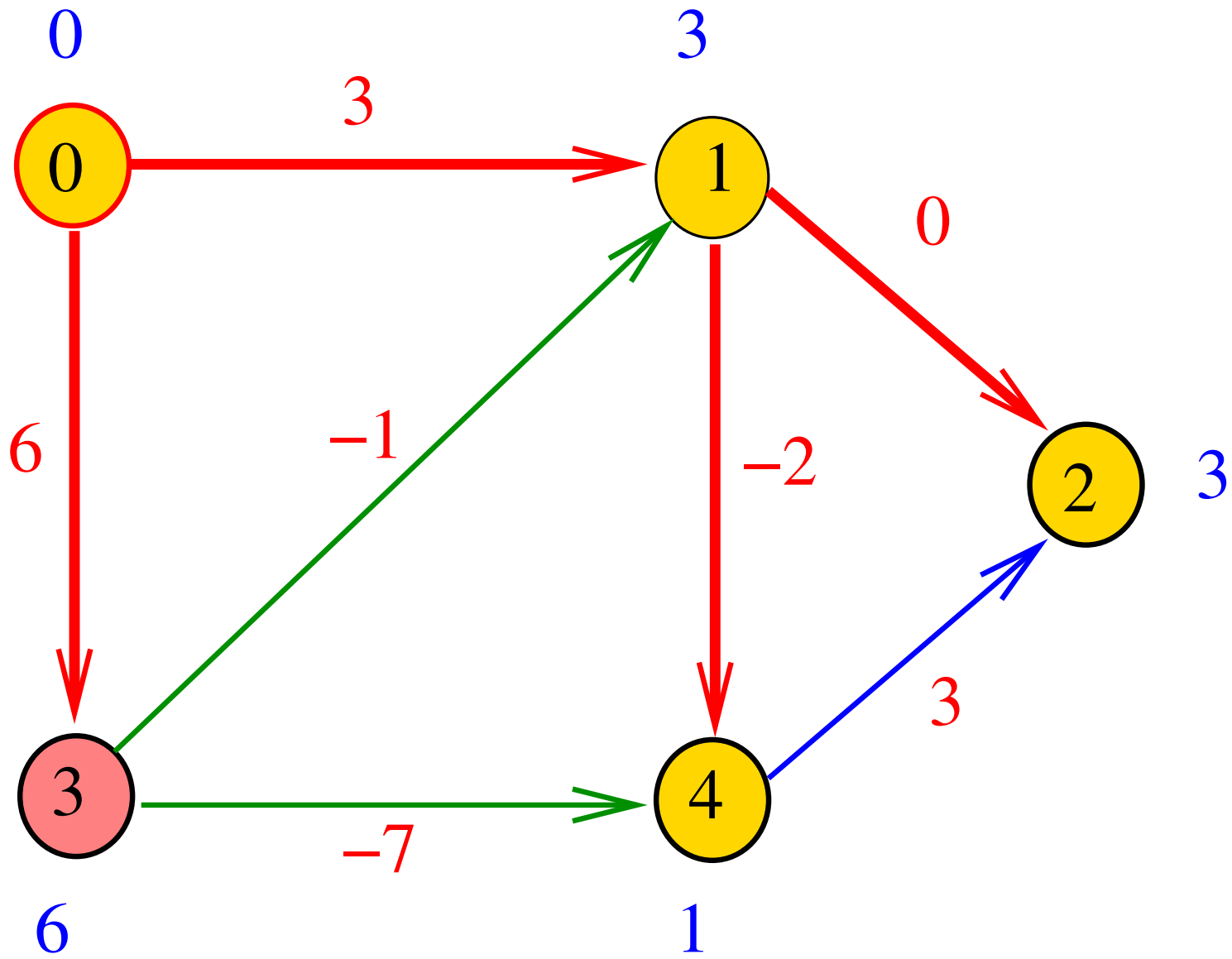
Apelemos para Dijkstra



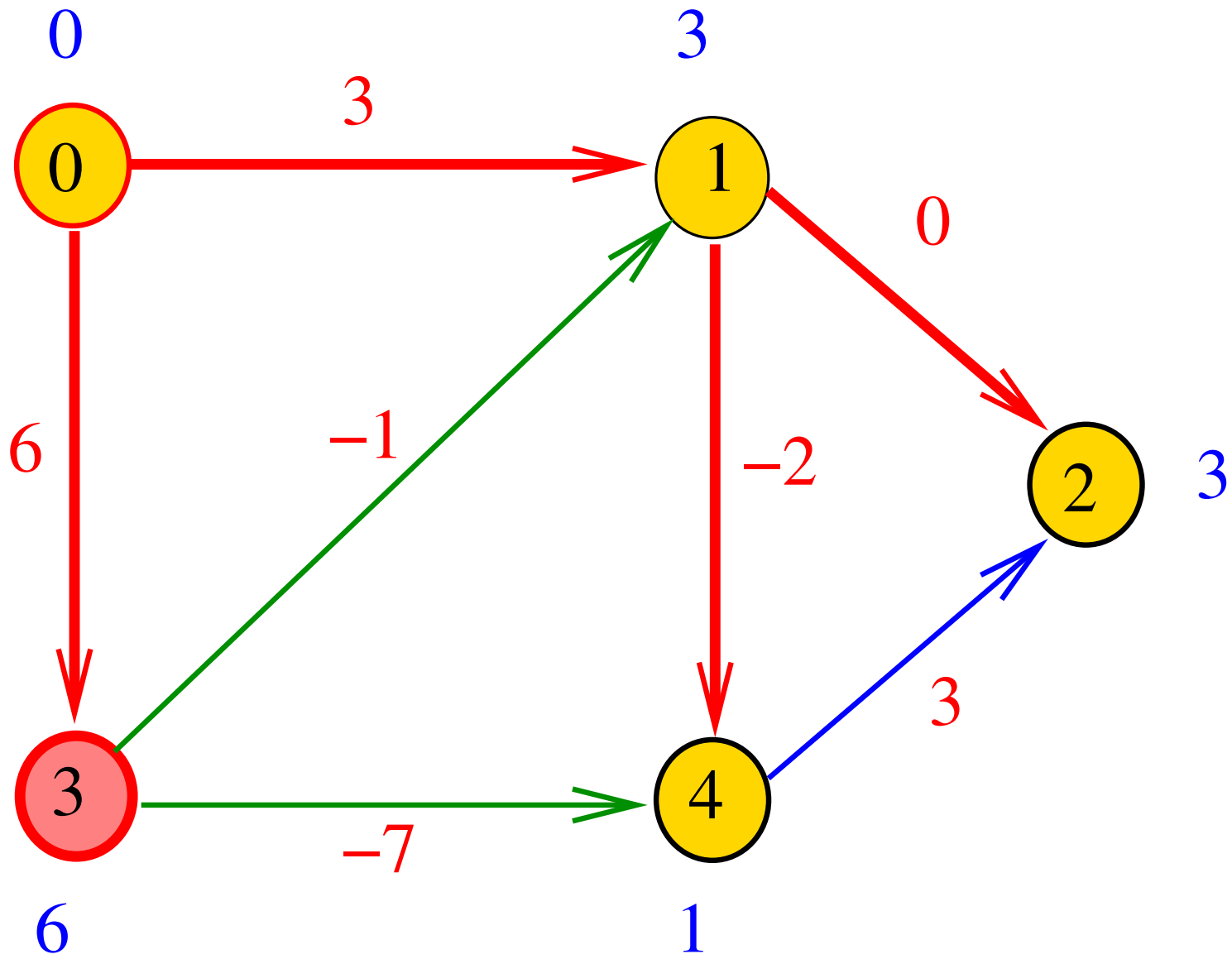
Apelemos para Dijkstra



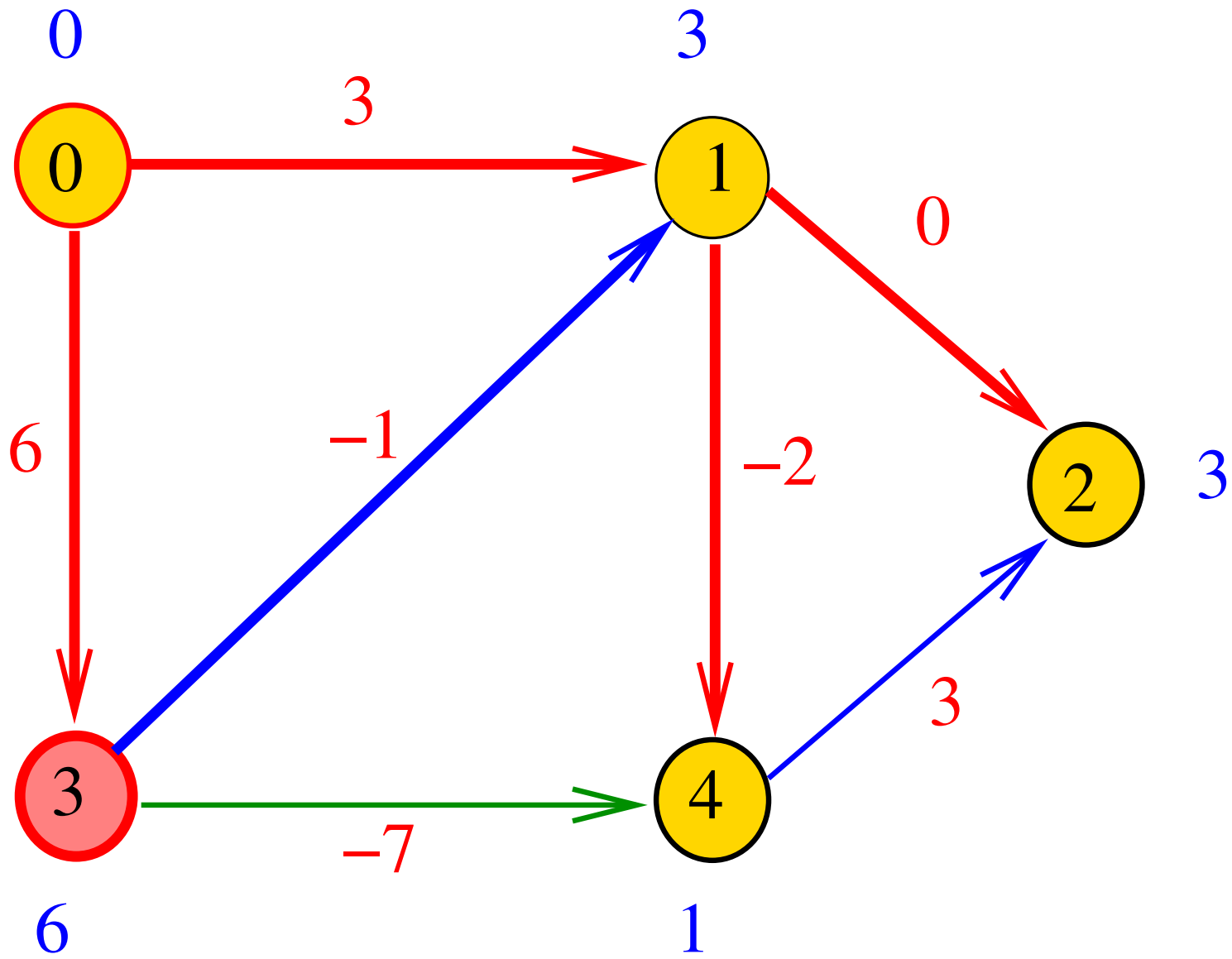
Apelemos para Dijkstra



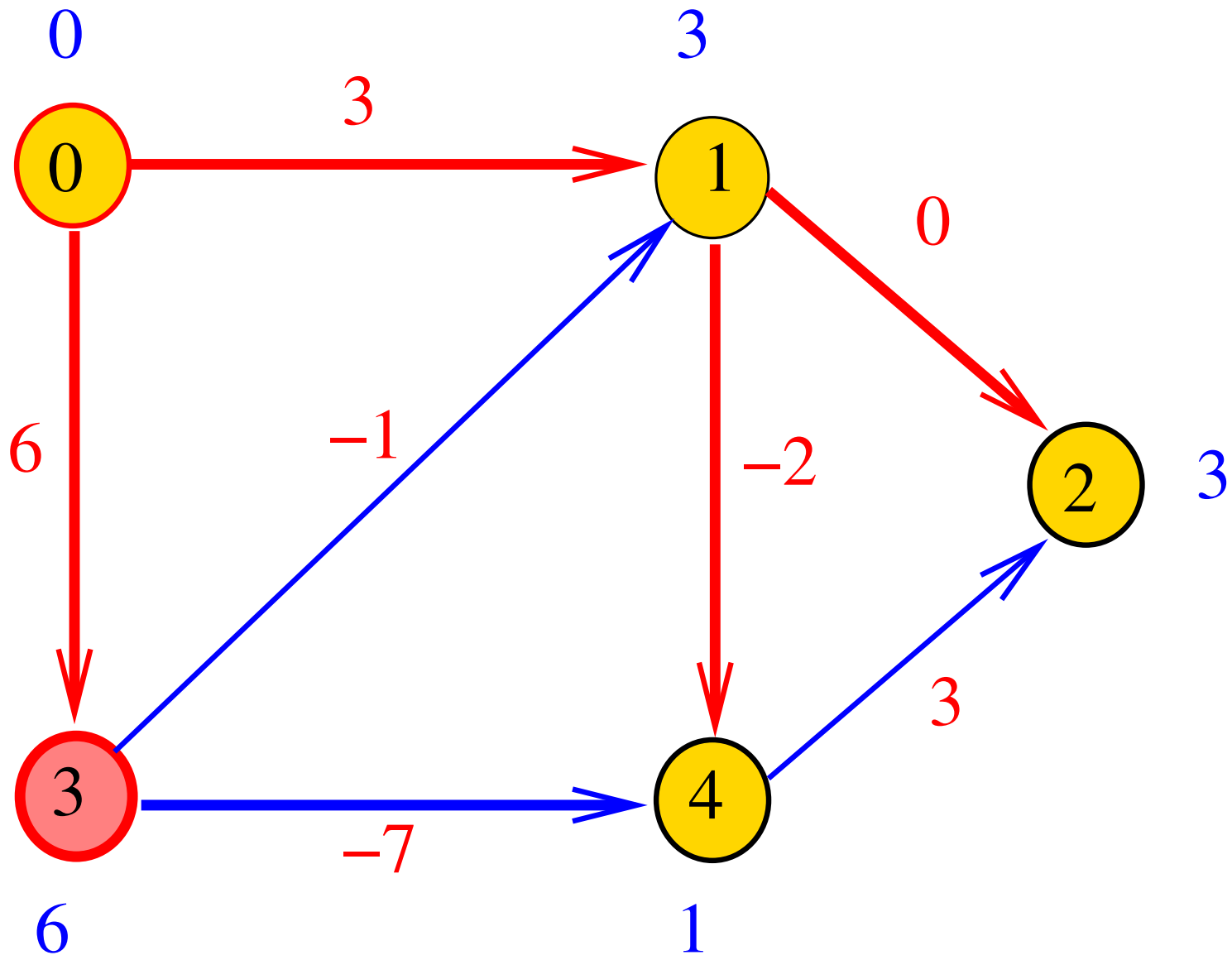
Apelemos para Dijkstra



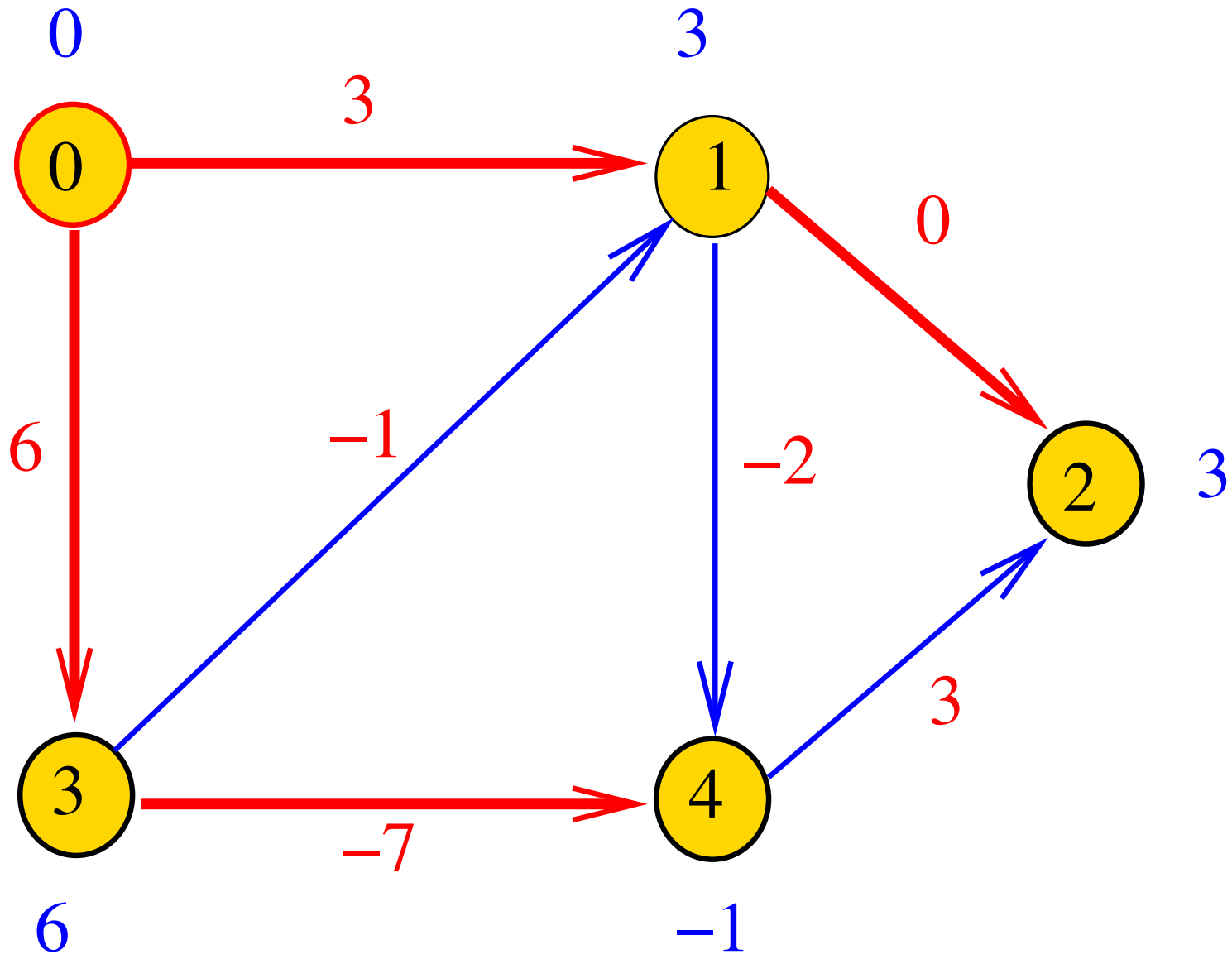
Apelemos para Dijkstra



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.

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 ...

Programação dinâmica

Propriedade (da subestrutura ótima)

Se G é um digrafo com custo 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 $0 \leq i \leq j \leq k$

$\texttt{custo}[v,w]$ = menor custo de uma caminho de v a w

Propriedade 1

O valor de $\texttt{custo}[s,t]$ é

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

Programação dinâmica

Propriedade (da subestrutura ótima)

Se G é um digrafo com custo 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

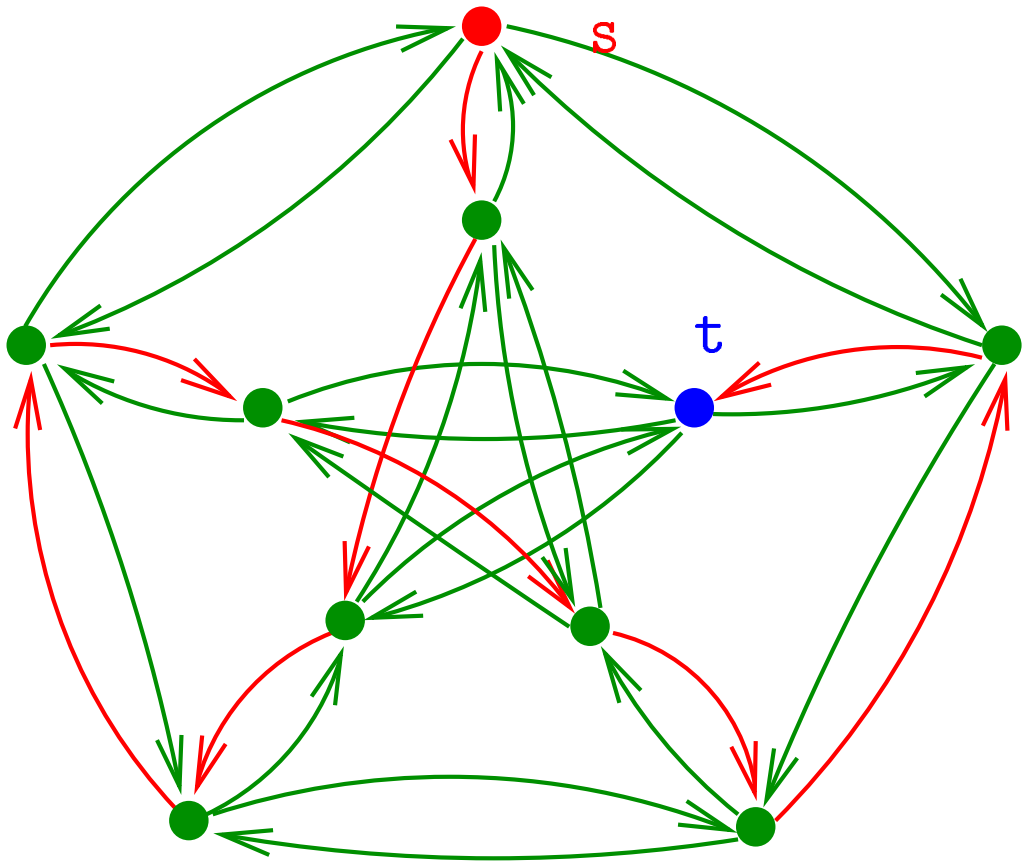
O valor de `custo`(s,t) é

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

Subestrutura ótima

todos custos $= -1$

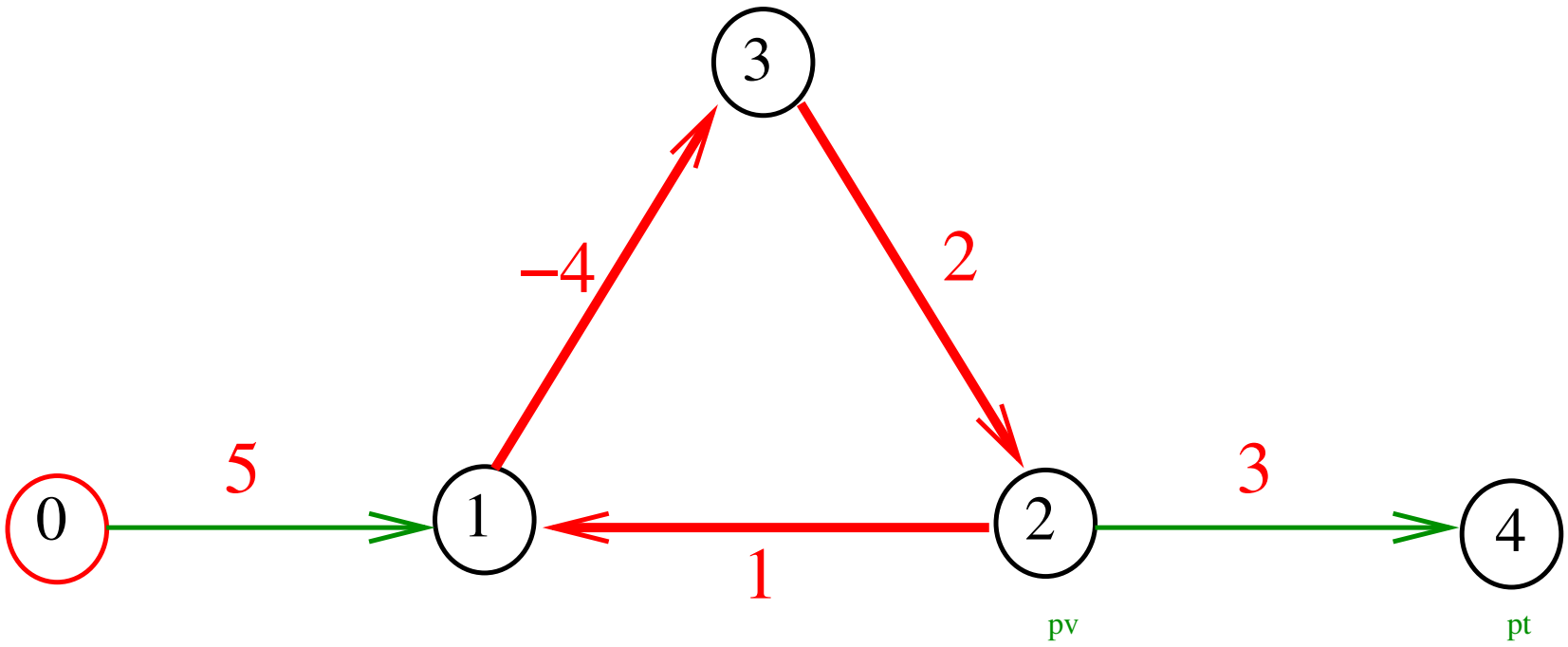
G



Não vale para digrafos com ciclos negativos

Ciclos negativos

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



Se o digrafo não possui ciclos negativos é possível encontrar caminhos mínimos.

AULA 17

Algoritmo de Bellman-Ford (cont.)

S 21.7

Programação dinâmica

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

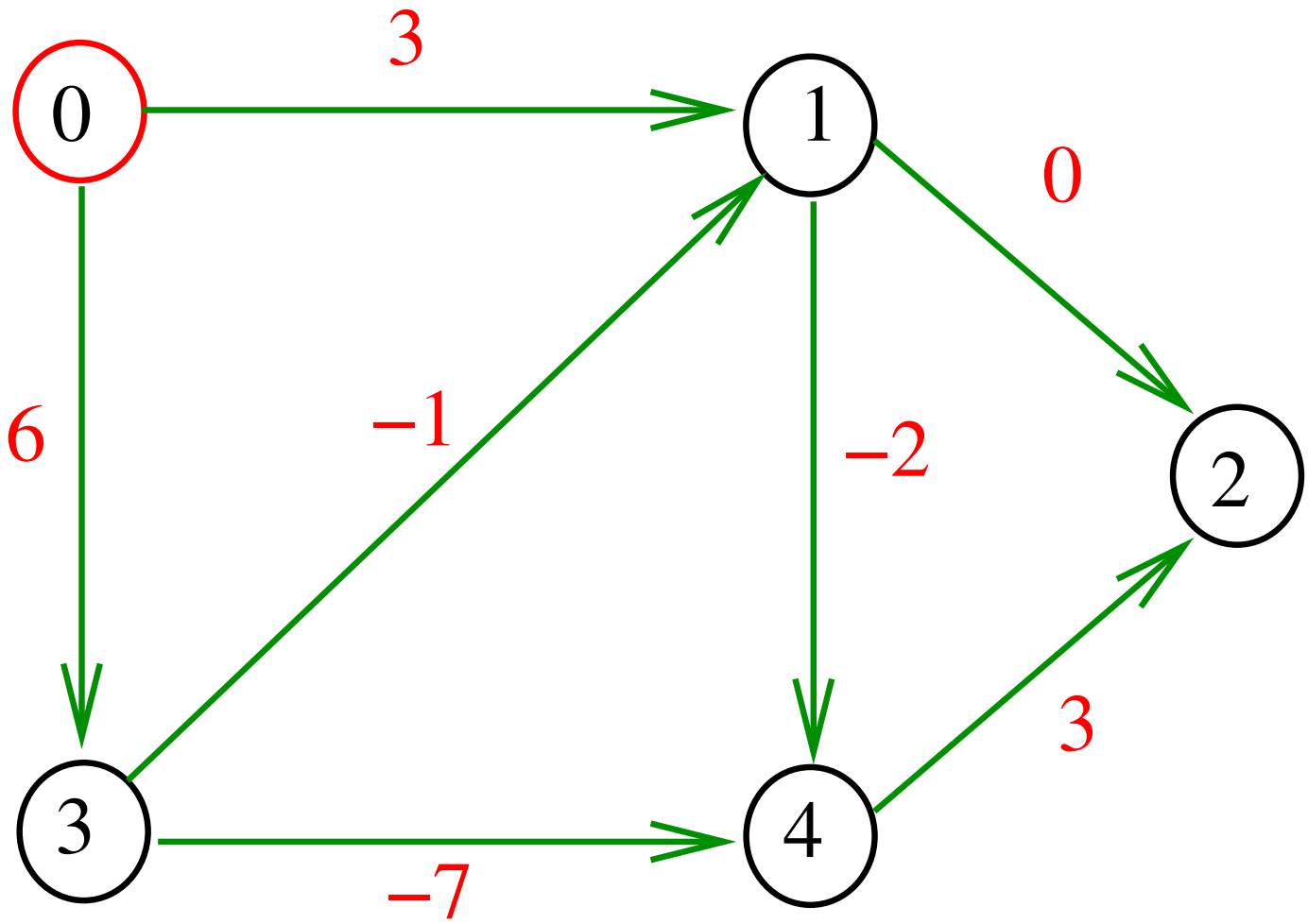
Recorrência:

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

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

```
void bellman_ford1(Digraph G, Vertex s){
 1    Vertex v, w; double d;
 2    for (v=0; v< G->V; v++)
 3         custo[0,v] = maxCST;
 4    custo[0,s] = 0;
 5    for (k=1; k<G->V; k++)
 6         for (w=0; w<G->V; w++){
 7              custo[k,w] = custo[k-1,w];
 8              for (v=0; v<G->V; v++){
 9                   d=custo[k-1,v]+G->adj[v,w];
10                   if (custo[k,w] > d)
11                        custo[k,w] = d;
                }
          }
}
```

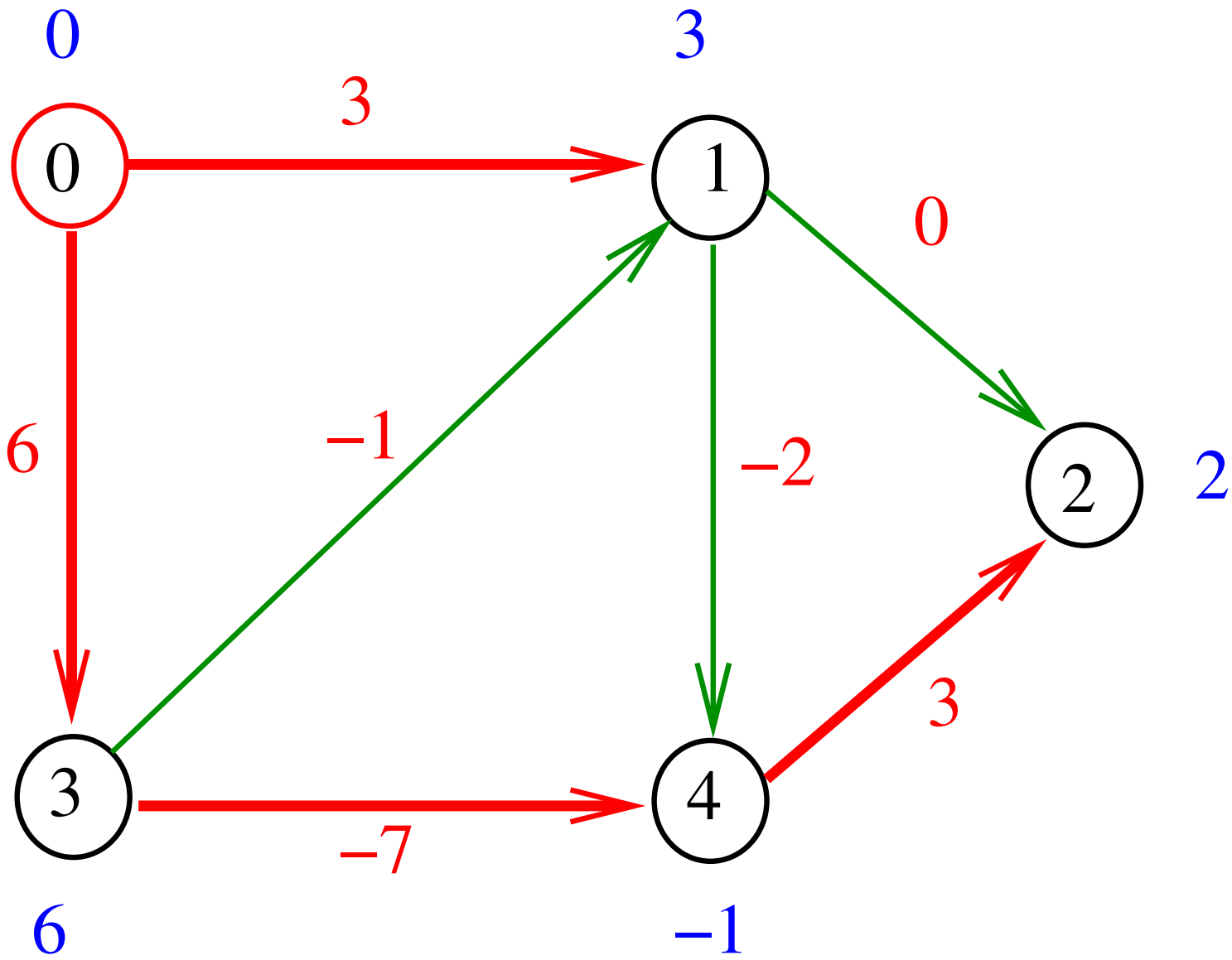
Exemplo



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



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 karcos.

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 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 $\mathtt{P}'$ tem no máximo $\mathtt{V}$-1 arcos e portanto

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

Logo $\mathtt{C}$ 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`.

Tabela da programação dinâmica

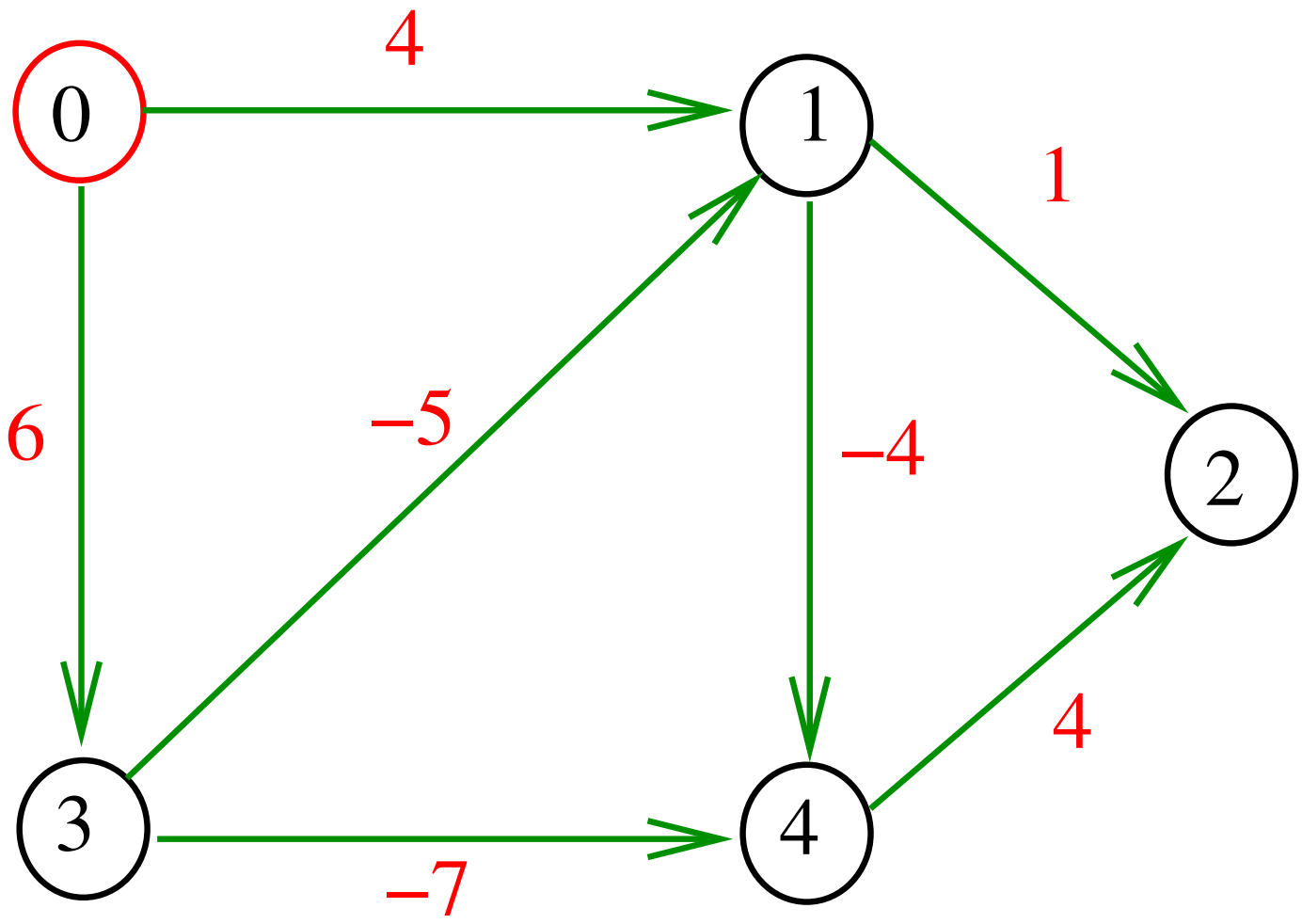
k \ v	1	2	3	4	5	s	7	8
0	$*$	$*$	$*$	$*$	$*$	0	$*$	$*$
1						0		
2						0		
3	$\star$	$\star$	$\star$	$\star$	$\star$	0	$\star$	$\star$
4				??		0		
5						0		
6						0		
7						0		

FIFO-Bellman-Ford

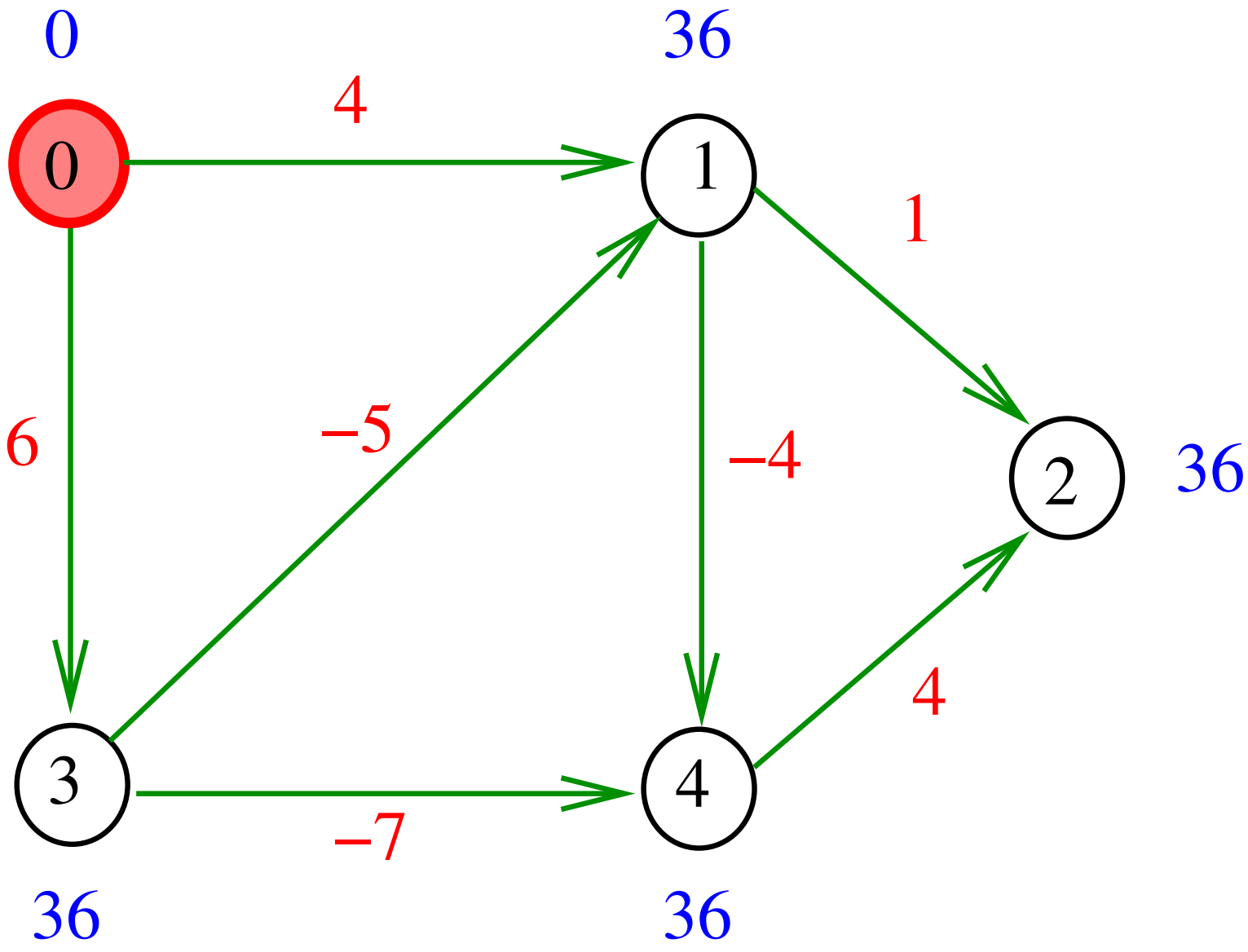
O algoritmo de Bellman e Ford pode ser divido em **passos**

um passo para cada valor de $k(=0,1,2,\ldots)$.

FIFO-Bellman-Ford

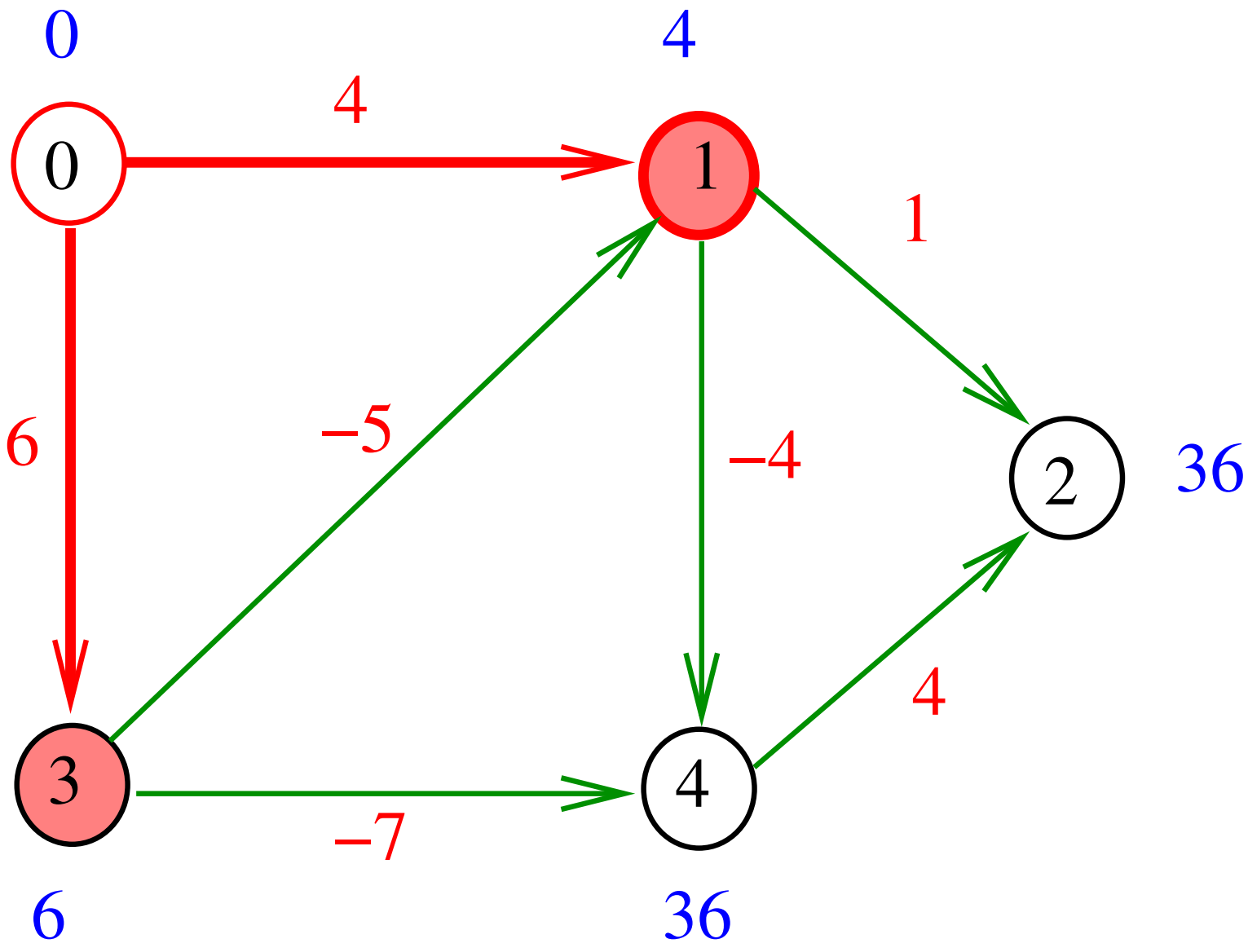


FIFO-Bellman-Ford



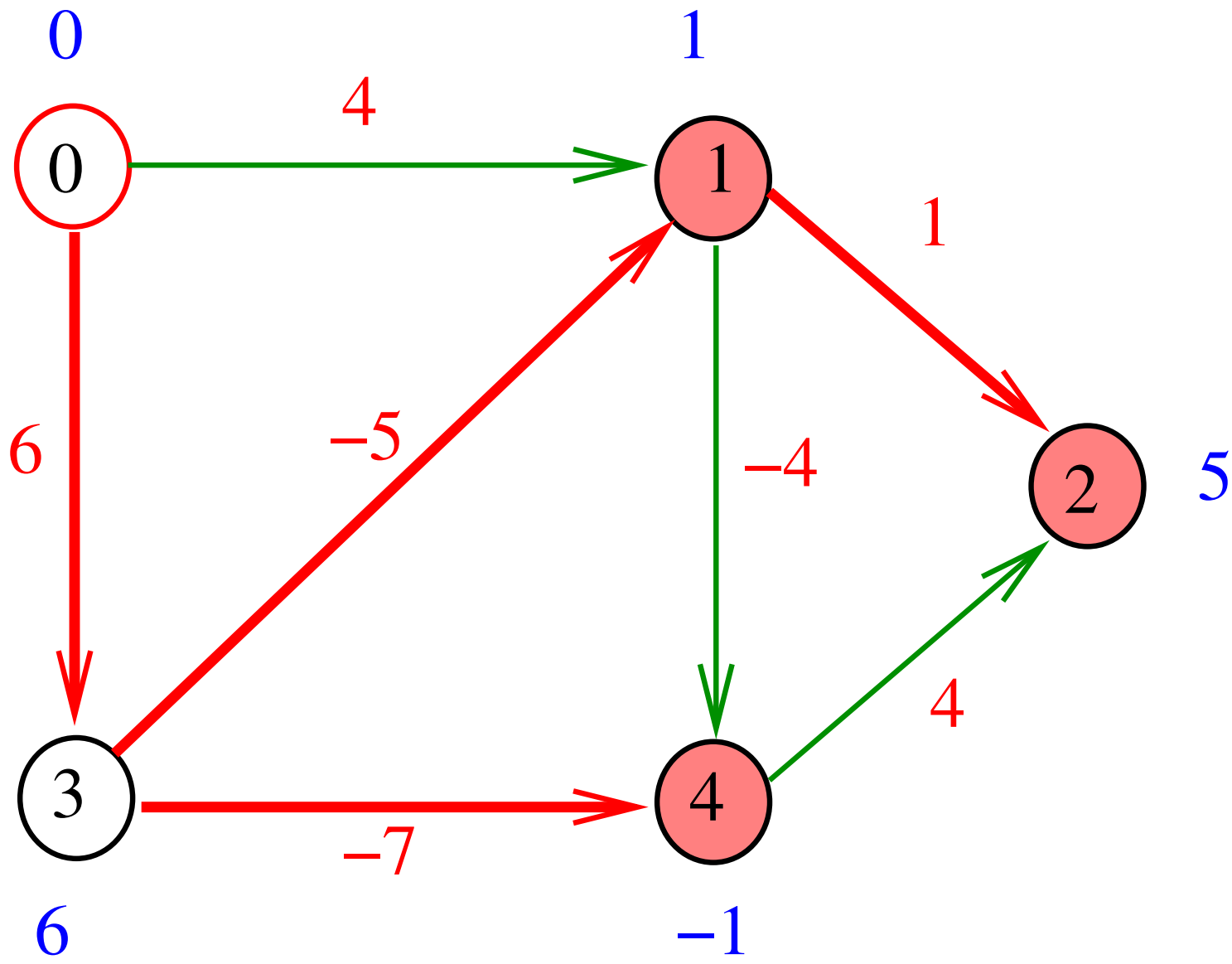
Fim do passo k=0

FIFO-Bellman-Ford



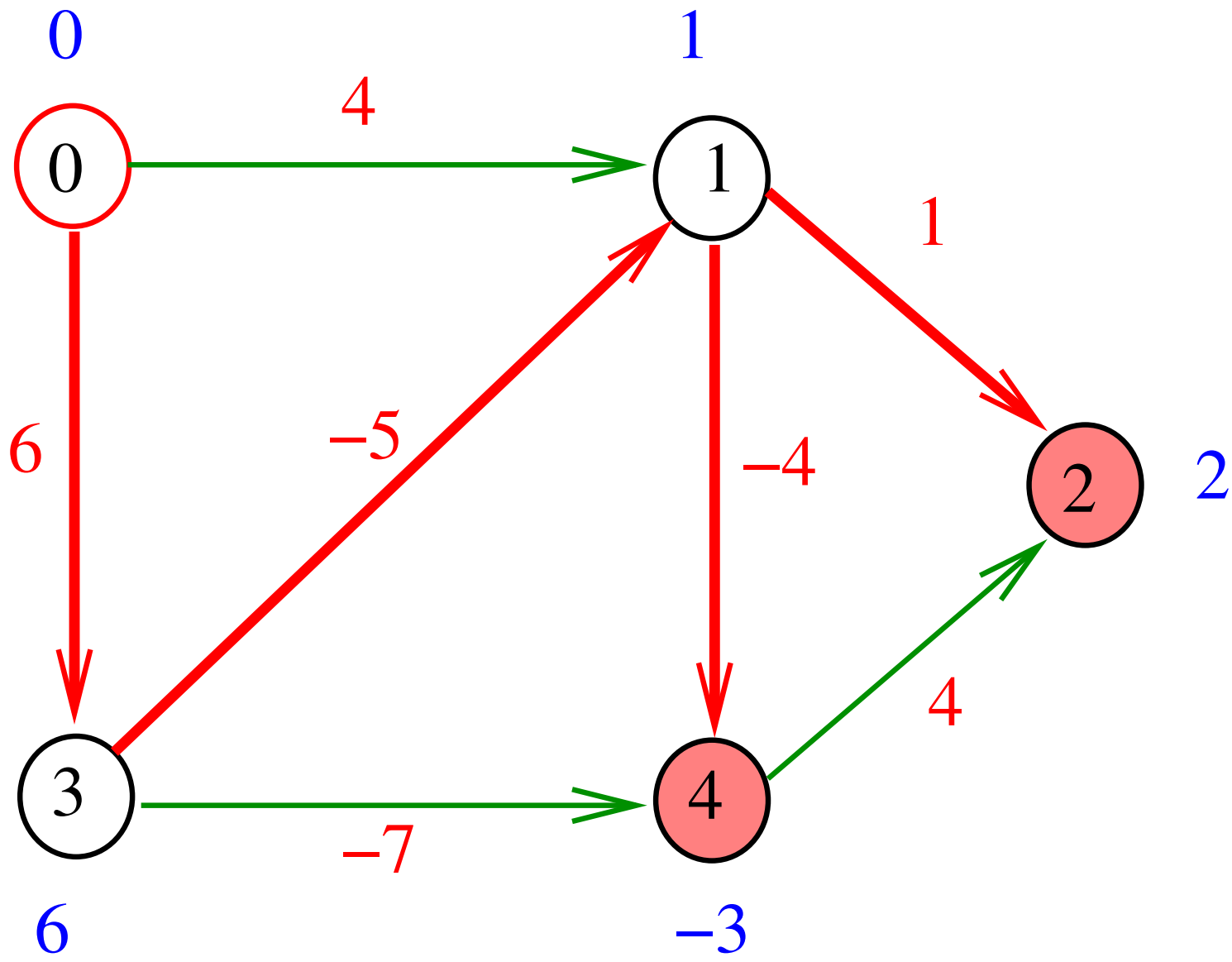
Fim do passo k=1

FIFO-Bellman-Ford



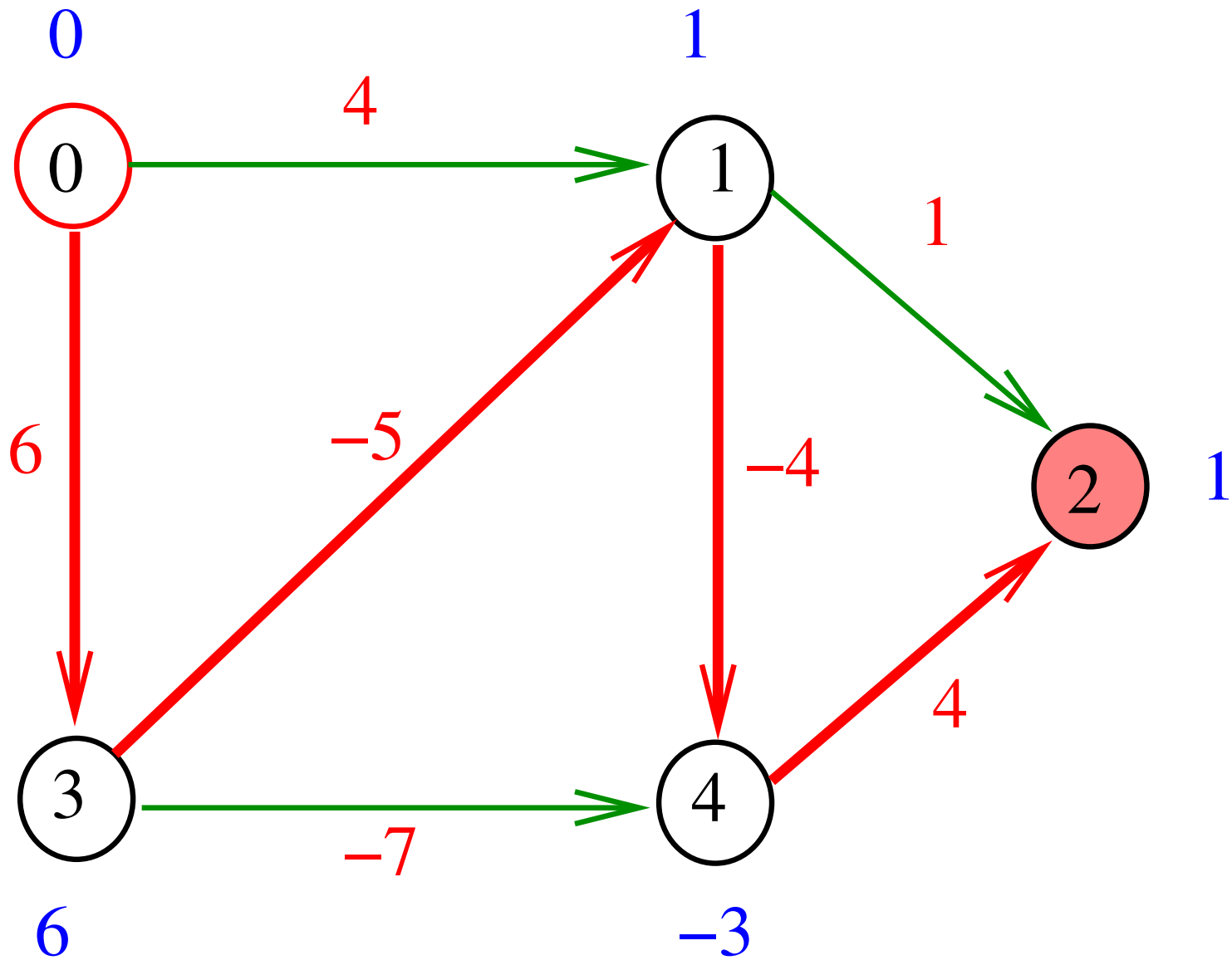
Fim do passo k=2

FIFO-Bellman-Ford



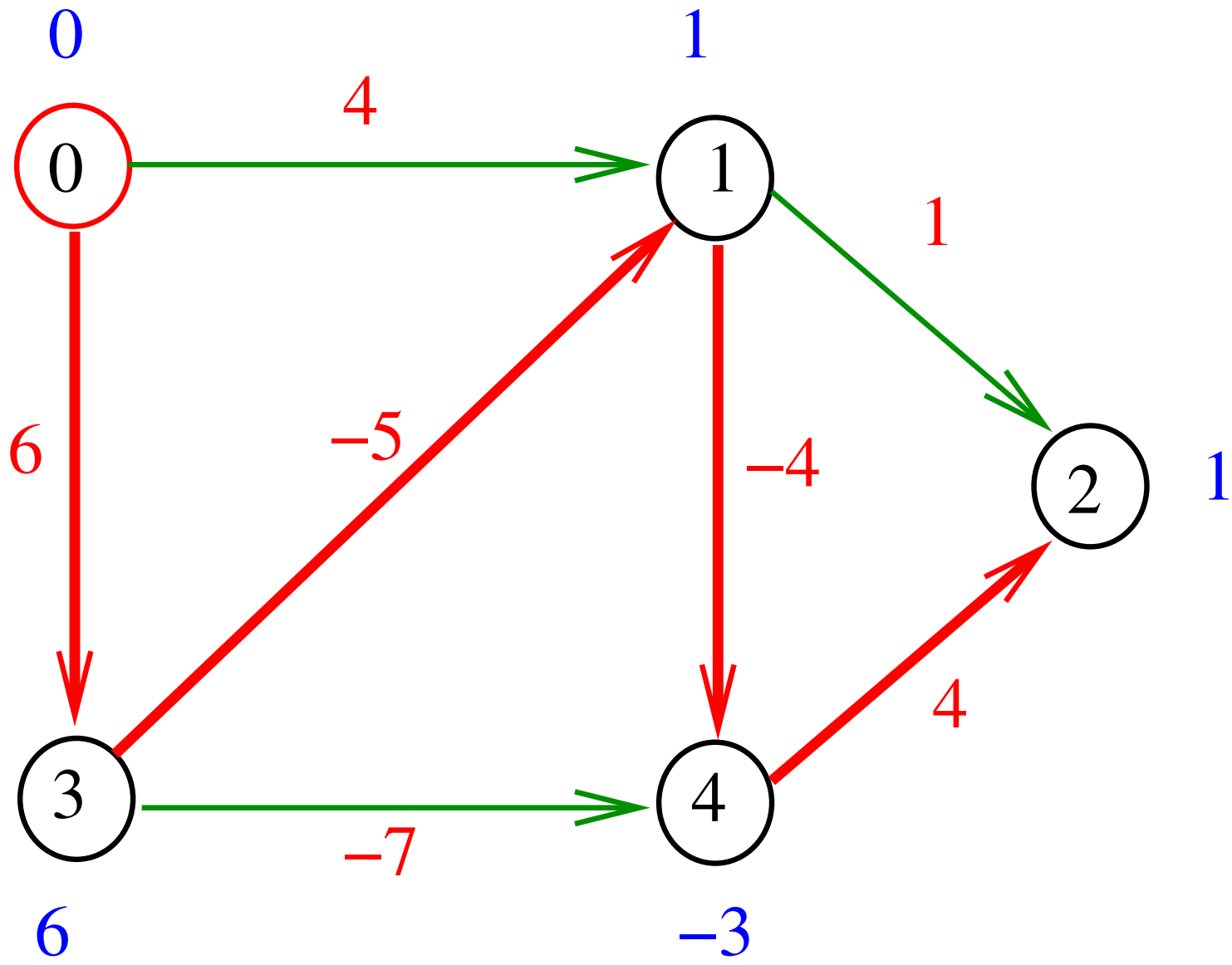
Fim do passo k=3

FIFO-Bellman-Ford



Fim do passo k=4

FIFO-Bellman-Ford



Fim do passo k=5