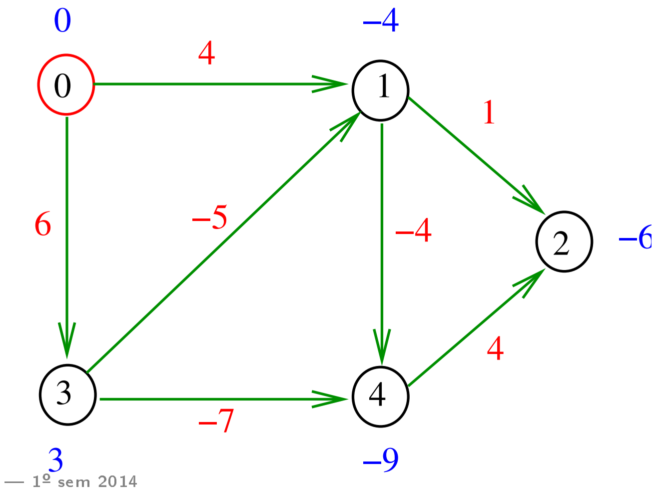
Mais Potenciais

Potenciais

Um **potencial** é um vetor y indexado pelos vértices do digrafo tal que

$\texttt{y[w]} - \texttt{y[v]} \leq \texttt{G->adj[v][w]}$ para todo arco v-w

Exemplo:

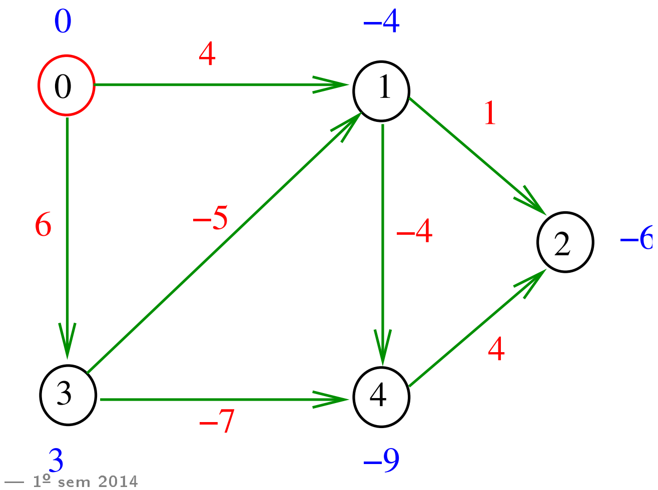


Potenciais

Um **potencial** é um vetor y indexado pelos vértices do digrafo tal que

$\texttt{y[w]} \leq \texttt{y[v]} + \texttt{G->adj[v][w]}$ para todo arco v-w

Exemplo:

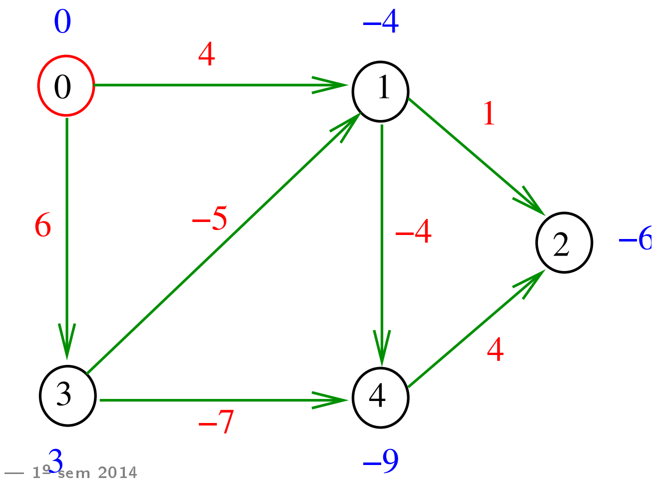


Propriedade dos potencias

Lema da dualidade. Se P é um caminho de s a t e y é um potencial, então

$$\texttt{custo(P)} \geq \texttt{y[t]} - \texttt{y[s]}$$

Exemplo:

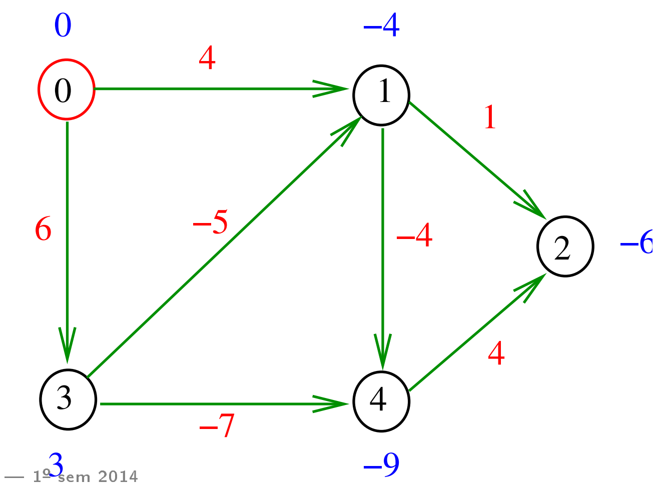


Propriedade dos potencias

Lema da dualidade. Se P é um caminho de s a t e y é um potencial, então

$$\texttt{custo}(P) + y[s] \geq y[t]$$

Exemplo:



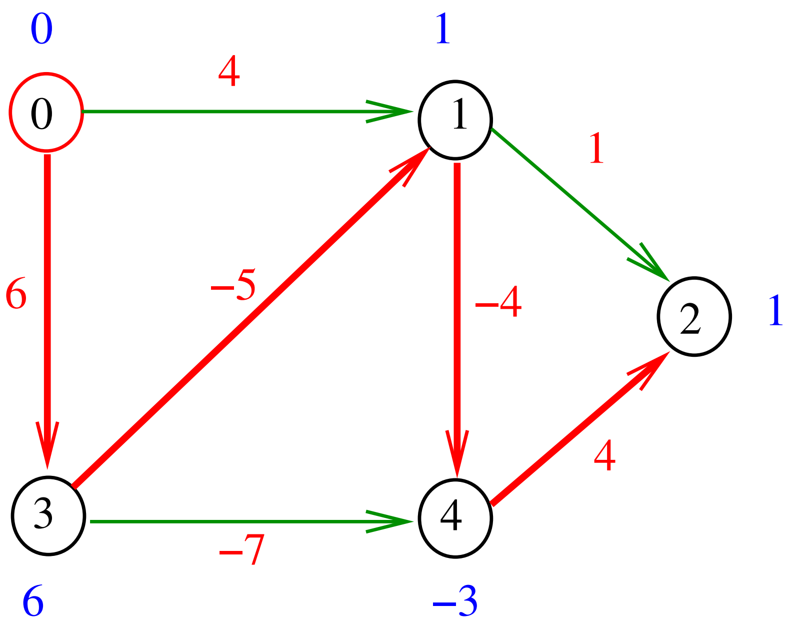
Conseqüência

Se P é um caminho de s a t e y é um potencial tais que

$$\texttt{custo}(P) = y[t] - y[s],$$

então P é um caminho **mínimo** e y é um potencial tal que $y[t] - y[s]$ é **máximo**

Exemplo



Teorema da dualidade

Da propriedade dos -potenciais (lema da dualidade) e da correção de `dijkstra` concluímos o seguinte:

Se s e t são vértices de um digrafo com custo **não-negativos** nos arcos e t está ao alcance de s então

$$\min\{\texttt{custo}(P) : P \text{ é um } s\ t\text{-caminho}\}$$
$$= \max\{y[t] - y[s] : y \text{ é um potencial}\}.$$

Algoritmo Floyd-Warshall

S 21.3

Problema dos caminhos mínimos entre todos os pares

Problema: Dado um digrafo com custo nos arcos, determinar, para cada par de vértices s, t o custo de um caminho mínimo de s a t

Esse problema pode ser resolvido aplicando-se V vezes o algoritmo Bellman-Ford

O consumo de tempo dessa solução é $O(V^2 A)$.

Um algoritmo mais eficiente foi descrito por Floyd, baseado em uma idéia de Warshall.

O algoritmo supõe que o digrafo não tem ciclo negativo

Programação dinâmica

$0, 1, 2, \ldots, V-1$ = lista dos vértices do digrafo

custo[k][s][t] = menor custo de um caminho de s a t usando vértices internos em $\{0, 1, \ldots, k-1\}$

Recorrência:

$$\begin{aligned} \texttt{custo[0][s][t]} &= \texttt{G->adj[s][t]} \\ \texttt{custo[k][s][t]} &= \min\{\texttt{custo[k-1][s][t]}, \\ &\quad \texttt{custo[k-1][s][k-1]} + \texttt{custo[k-1][k-1][t]}\} \end{aligned}$$

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


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

Consumo de tempo

> O consumo de tempo da função `floyd_warshall1` é $O(V^3)$.


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


Relação invariante

No início de cada iteração da linha 5 vale que

cst[s][t] = custo[k][s][t] = o menor custo de um caminho de s a t usando vértices internos em $\{0, 1, \ldots, k-1\}$

Novo resumo

função	consumo de tempo	observação
`DAGmin`	$O(V + A)$	digrafos acíclicos custos arbitrários
`dijkstra`	$O(A \lg V)$	custos ≥ 0, min-heap
	$O(V^2)$	custos ≥ 0, fila
`bellman-ford`	$O(V^3)$ $O(VA)$	digrafos densos digrafos esparsos
`floyd-warshall`	$O(V^3)$	digrafos sem ciclos negativos

O problema SPT em digrafos com ciclos negativos é NP-difícil.