Clase 12 MA3705. 26 de octubre 2020.

Tabla

- Dijkstra (revisitado)
- Calcular distancia en todos los pares

Bellman Ford.

largos conservativos.

$O(nm)$ $O(n(n+m))$

Dijkstra.

largos ≥ 0.

$O(m+n\log n)$

Dijkstra revisitado

Algoritmo de Dijkstra (Dijkstra 1959):

Entrada: $G = (V, E)$ dirigido, $s \in V$, $\ell \colon E \to \mathbb{R}^+$

para $v \in V$ **hacer** $d(s, v) = +\infty$. $T(v) \leftarrow \infty$. $\textsc{Padre}(v) \leftarrow \textsc{Null}$

$\textsc{No-Visitados} \leftarrow V$.

$T(s) \leftarrow 0$.

mientras $\textsc{Tmin} \leftarrow \min\limits_{v \in \textsc{No-Visitados}} T(v) < +\infty$ **hacer**

    Elegir $v \in \textsc{No-Visitados}$ con $T(v) = \textsc{Tmin}$.

    $d(s, v) \leftarrow T(v)$

    $\textsc{No-Visitados} \leftarrow \textsc{No-Visitados} - v$

    para $w \in N^+(v) \cap \textsc{No-visitado}$ **hacer**

        si $T(v) + \ell(vw) < T(w)$ **entonces**

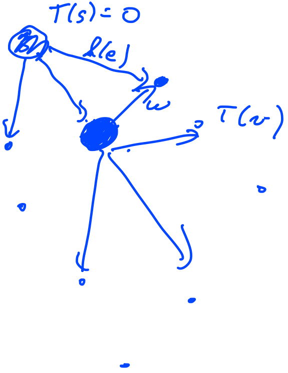
            $T(w) \leftarrow T(v) + \ell(vw), \textsc{Padre}(w) \leftarrow v$

        fin

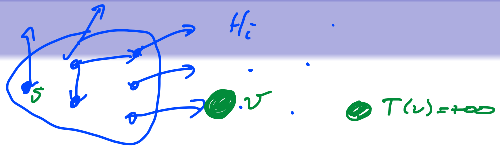
    fin

fin

devolver $d, \textsc{Padre}$



Correctitud de Dijkstra



Inicialización

$(\forall v \in V) \quad d(s,v) \leftarrow +\infty,\ T_1(v) \leftarrow +\infty.$
$\textsc{NoVis} \leftarrow V.\ T_1(s) \leftarrow 0.$

Iteración $i+1$

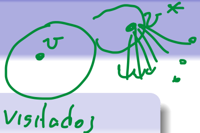
$T_{i+1} \equiv T_i$
$\textsc{Tmin} \leftarrow \min_{v \in \textsc{NoVis}} T_i(v)$
si $\textsc{Tmin} = +\infty$ **entonces** terminar
Elegir $v^* \in \textsc{NoVis}$ con $T_i(v^*) = \textsc{Tmin}$.
$\textsc{NoVis} \leftarrow \textsc{NoVis} - v^*$
$d(s,v^*) \leftarrow T_i(v^*)$
$(\forall w \in N^+(v^*) \cap \textsc{NoVis})$
$T_{i+1}(w) \leftarrow \min(T_i(w), T_i(v^*) + \ell(vw))$

Teorema: Si H_i es el grafo de los arcos con colas visitadas al principio de la iteración i

$\forall v$ visitado:
$d(s,v) = \text{dist}_G(s,v) = \text{dist}_{H_i}(s,v)$
$\forall v$ No visitado:
$T(v) = \text{dist}_{H_i}(s,v)$

Iteración 1: $H_1 = (V, \emptyset)$

$$T(v) = \begin{cases} 0 & v = s \\ +\infty & v \neq s. \end{cases}$$

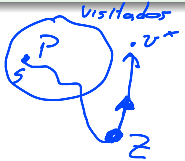


Teorema

$\forall v$ visitado:
$d(s,v) = \text{dist}_G(s,v) = \text{dist}_{H_i}(s,v)$
$\forall v$ No visitado: $T(v) = \text{dist}_{H_i}(s,v)$

Cambios iteración $i+1$

$d(s,v^*) = T_i(v^*)$
$(\forall w \in N^+(v^*) \cap \text{NoVis})$
$T_{i+1}(w) \leftarrow \text{mín}(T_i(w), T_i(v^*) + \ell(vw))$



Demostración:

1. Si v visitado en $\leq i$.

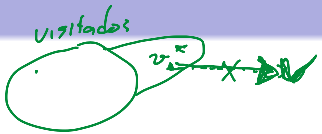
$d(s,v) = \cancel{T(v)} = \boxed{\text{dist}_{H_i}(s,v)} \underset{\le}{\overset{=}{\geq}} \text{dist}_{H_{i+1}}(s,v) \overset{=}{\geq} \text{dist}_G(s,v) = \boxed{\text{dist}_{H_i}(s,v)}.$

2. Si $v = v^*$

$d(s,v^*) = T_i(v^*) = \text{dist}_{H_i}(s,v^*) = \text{dist}_{H_{i+1}}(s,v^*) \geq \text{dist}_G(s,v^*) = \ell(P) \geq T_i(v^*).$
con P camino mínimo de s a v^*.
Si P no usa arcos fuera de H_i, entonces $\ell(P) = \text{dist}_{H_i}(s,v^*)$ ✓
Si P usa arcos fuera de H_i, entonces .. $\ell(P) \geq \underbrace{\ell(sPz)}_{\in H_i} + \underbrace{\ell(zPv)}_{\geq 0} \geq \text{dist}_{H_i}(s,z) = T_i(z) \geq T_i(v^*)$

Sea z un nodo no visitado, dentro de P

largos ≥ 0



Teorema

$\forall v$ visitado:
$d(s,v) = \text{dist}_G(s,v) = \text{dist}_{H_i}(s,v)$
$\forall v$ No visitado: $T(v) = \text{dist}_{H_i}(s,v)$

Cambios iteración $i+1$

$d(s,v^*) = T_i(v^*)$
$(\forall w \in N^+(v^*) \cap \textsc{NoVis})$
$\quad T_{i+1}(w) \leftarrow \min(T_i(w), T_i(v^*) + \ell(vw))$

Demostración:

3. Si v no visitado, $v \notin N^+(v^*)$

$$T_{i+1}(v) = T_i(v) = \text{dist}_{H_i}(s,v) = \text{dist}_{H_{i+1}}(s,v) \checkmark$$

4. Si v no visitado, $v \in N^+(v^*)$

$$T_{i+1}(v) = \min(T_i(v), T_i(v^*) + \ell(v^*v)) = \min(\text{dist}_{H_i}(s,v), \boxed{d(s,v^*) + \ell(v^*v)}) \geq \text{dist}_{H_{i+1}}(s,v)$$

$\ell(P)$" ↑ si no usa v^* ↑ si usa v^*

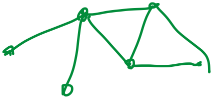
P camino mínimo en H_{i+1} de s a v

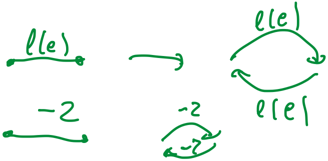
Resumen de Caminos mínimos desde un nodo s en grafos dirigidos

Prog. dinámica

Función de largo	Algoritmo	Complejidad
Cualquiera en DAG	Orden Topológico + PD	$O(n+m)$
Unitaria	BFS	$O(m+n)$
Enteros entre 1 y M	BFS	$O(M(m+n))$
No negativos	Dijkstra	$O(m+n\log n)$
Conservativos	Bellman Ford	$O(n(n+m))$

¡Ojo: Estos algoritmos también funcionan en grafos no dirigidos (basta bidirigir aristas)!





Distancias en todos los pares

Idea 1: Correr distancia de fuente única desde cada nodo.

Función de largo	Algoritmo	Complejidad
Cualquiera en DAG	Orden Topológico + PD	$O(n\cdot(n+m))$
Unitaria	BFS	$O(n\cdot(n+m))$
Enteros entre 1 y M	BFS	$O(n\cdot M(n+m))$
No negativos	Dijkstra	$\boxed{O(n(m+n\log n))}$
Conservativos	Bellman Ford	$O(n\cdot \underbrace{m(n+m)}))$

Para largos conservativos en grafos densos $(m = \Theta(n^2))$, esto es: $O(n^4)$
que es demasiado lento.

Bellman Ford usa (llamando $D^{(k)}(a,b) = d_{\leq k}(a,b)$)

$$D^{(i+1)}(s,t) = \min\left(D^{(i)}(s,t), \min_{u \in N^-(t)} D^{(i)}(s,u) + \ell(u,t)\right)$$

Esto equivale a:

$D^{(1)}(t,t) = 0$

$$\begin{aligned} D^{(i+1)}(s,t) &= \min\left(D^{(i)}(s,t), \min_{u \in N^-(t)} D^{(i)}(s,u) + D^{(1)}(u,t)\right) \\ &= \min_{u \in N^-(t)+t} D^{(i)}(s,u) + D^{(1)}(u,t) \\ &= \min_{u \in V} D^{(i)}(s,u) + D^{(1)}(u,t). \end{aligned}$$

$D^{(i+1)} = D^{(i)} \odot D^{(1)}$

$D^{(1)}(v,t) = +\infty$

Producto de matrices en semianillo $(\mathbb{R}_+^{\infty}, \min, +)$

Si A, B son matrices en $\mathbb{F}^{V\times V}$ entonces

$$(A\cdot B)(s,t) = \sum_{u\in V} A(s,u)\cdot B(u,t)$$

Consideremos ahora matrices en $(\mathbb{R}_+^{+\infty})^{V\times V}$

$$(A\odot B)(s,t) = \min_{u\in V}(A(s,u)+B(u,t))$$

¡El producto recién definido es asociativo!
(No es anillo pues la primera operación $(\min)$ no tiene inversa).

$$(A\odot B)\odot C = A\odot(B\odot C)$$

¡Calcular distancia es multiplicación de matrices!

$$D^{(n-1)} = D^{(n)} = D^{(n+1)} = D^{(n+2)} \dots$$

Calcular todas las distancias

Entrada: $G = (V, E)$ dirigido, $\ell \colon E \to \mathbb{R}$ conservativo

$$D^{(1)}(a,b) = \begin{cases} 0 & \text{si } a = b \\ \ell(ab) & \text{si } ab \in E \\ +\infty & \text{en otro caso} \end{cases} \quad \leftarrow \quad \Big] \; O(n^2) \Big]$$

para $k = 2, \dots, n-1$ **hacer**
| $\underline{D^{(k)}} = \underline{D^{(k-1)}} \odot \underline{D^{(1)}}$ $\Big]$ $O(n^3)$
fin
devolver $D^{(n-1)}$

Complejidad: $O(n^2) + O(n \cdot n^3) = O(n^4)$

Calcular todas las distancias

Entrada: $G = (V, E)$ dirigido, $\ell \colon E \to \mathbb{R}$ conservativo
Sea M el menor entero tal que $2^M \geq n - 1$.

$$D^{(1)}(a, b) = \begin{cases} 0 & \text{si } a = b \\ \ell(ab) & \text{si } ab \in E \\ +\infty & \text{en otro caso} \end{cases} \quad O(n^2)$$

para $k = 1, \dots, M$ **hacer**
  $D^{(2^k)} = D^{(2^{k-1})} \odot D^{(2^{k-1})}$ $[O(n^3)]$
fin
devolver $D^{(2^M)}$

$M = \Theta(\ln n)$

Complejidad
$O(M \cdot n^3)$
$= O(n^3 \log n)$

Propuesto: Modificar el algoritmo anterior para que guarde matrices de (argmins) $R^{(2^k)}_{k=1\dots M}$. Suponga acceso directo a todas las matrices calculadas y muestre como calcular un s-t camino mínimo en tiempo $O(n)$

Roy (1959), Floyd(1952), Warshall(1962) se dieron cuenta que la idea de iterativamente aumentar el conjunto de arcos usables es más eficiente que la multiplicación de matrices.

Para $G = (V, E)$ dirigido, ℓ conservativo. Enumeremos los vértices como $\underbrace{v_1, \ldots, v_n}$.

$\{v_1, \ldots, v_i\}$

$$D(u, v, i) = \min\{\ell(P) \colon P \text{ es } u\text{-}v \text{ camino con vértices internos en } V_i\}$$

Con esto:

$$D(u, v, 0) = D^{(1)}(u, v)$$

$$D(u, v, i+1) = \min\Big(D(u,v,i),\ D(u, v_{i+1}, i) + D(v_{i+1}, v, i) \Big)$$

no usa v_{i+1}

si usa v_{i+1}

Algoritmo de (Roy)-Floyd-Warshall

Entrada: $G = (V, E)$ dirigido, $\ell \colon E \to \mathbb{R}$ conservativo

$$D(a,b,1) = \begin{cases} 0 & \text{si } a = b \\ \ell(ab) & \text{si } ab \in E \\ +\infty & \text{en otro caso} \end{cases}$$

para $k = 2, \ldots, n, a \in V, b \in V$ **hacer**

$D(a,b,k) =$
$\min(D(a,b,k-1), D(a,v_k,k-1) + D(v_k,b,k-1))$ $O(1)$

fin

devolver $D(a,b,n)$

Complejidad
$O(n^3)$

Propuesto: Suponga acceso directo al arreglo D y muestre como calcular un s-t camino mínimo en tiempo $O(n)$. Más aún, muestre como calcular un árbol de distancias desde s en tiempo $O(n+m)$.

¿Cómo determinar si un largo es conservativo?

1. Bellman-Ford: Existe un ciclo negativo alcanzable desde s si y solo si no hay desigualdad triangular en $d_{\leq n-1}$. Es decir, existen $vw \in E$ tal que $d_{\leq n-1}(s,v) > d_{\leq n-1}(s,w) + \ell(w,v)$
2. Floyd-Warshall: Existe ciclo negativo si y solo si $D(a,a,n) < 0$ para algún $a \in V$.

Propuestos:

1. Modificar BF para que devuelva el ciclo negativo encontrado.
2. Modificar FW para que devuelva el ciclo negativo encontrado.