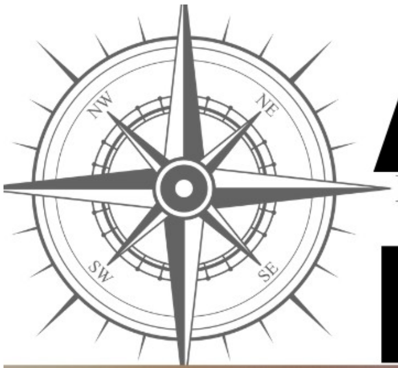
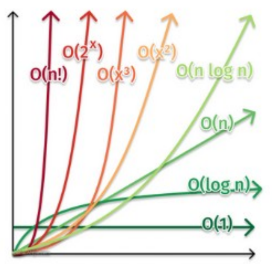


All pairs shortest paths



Introducción

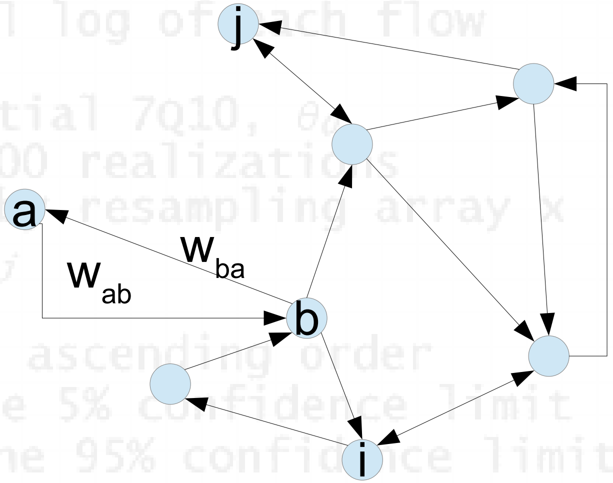
Teoría de Algoritmos
Víctor Podberezski
vpodberezski@fi.uba.ar

por Vpode

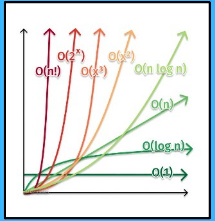
Problema

- Sea
  - G=(V,E) un grafo ponderado dirigido
  - Sin ciclos negativos

  Donde
  - cada vértice e=(a,b) ∈ E tiene un peso w_{ab}
- Queremos saber
  - para cada par i,j ∈ V el camino mínimo entre ellos

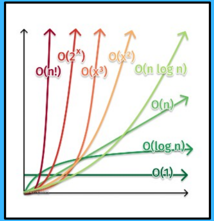


Aplicaciones



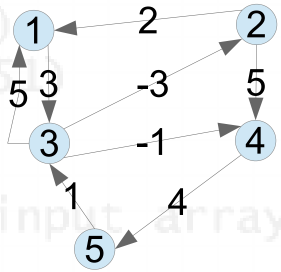
- Sistema de gestión de trafico
- Diseño de circuitos electrónicos
- Procesamiento de imágenes
- Análisis de redes
- Robótica
- Análisis de ADN
- Compuestos químicos

Matriz de adyacencias



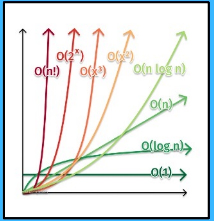
- Para representar el grafo usaremos
  - una matriz W de |V| x |V| adyacencias levemente modificada

$$w_{ij} = \begin{cases} 0 & \text{si } i=j \\ \text{peso arista de i a j} & \text{si } (i,j) \in E \\ \infty & \text{si } (i,j) \notin E \end{cases}$$



	1	2	3	4	5
1	0	∞	3	∞	∞
2	2	0	∞	∞	5
3	5	-3	0	-1	∞
4	∞	∞	∞	0	4
5	∞	∞	1	∞	0

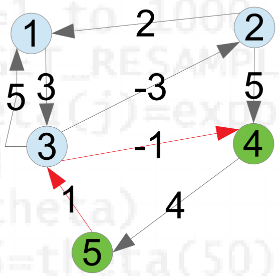
Salida de la ejecución del algoritmo

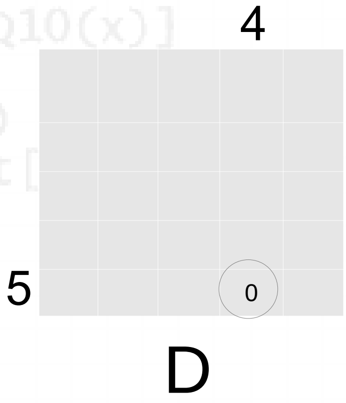


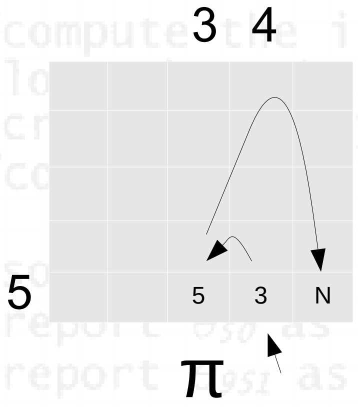
- Matriz “D” de distancias mínimas
  - tamaño |V| x |V|
  - d_{ij} contiene el peso del camino mínimo desde el vértice i al vertice j (∞ si no existe un camino)
- Matriz “π” de predecesores
  - tamaño |V| x |V|
  - π_{ij} el vértice predecesor j en el camino mínimo desde i (NULL si i=j o si no existe un camino mínimo entre i y j)

Ejemplo

- Camino mínimo de 5 a 4:

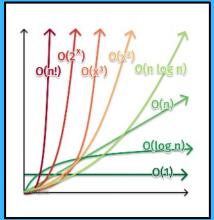






Camino p: 5 → 3 → 4
Distancia: 0

Pseudocódigo: Camino mínimo




```
camino_minimo(PI, i, j)
    Si i == j
        imprimir i
    Sino si PI[i,j] == NULL
        imprimir ‘no hay camino de i a j’
    Sino
        camino_minimo(PI,i,PI[i,j])
        imprimir j
```

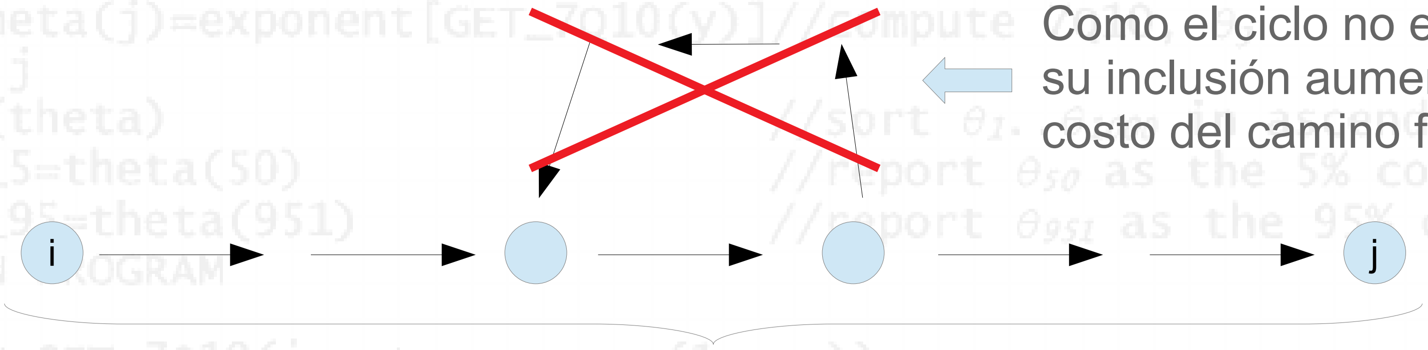
Subcaminos mínimos

- Todo subcamino de un camino mínimo
  - Corresponde a su vez a un camino mínimo

  - *Por el absurdo: supongamos que existe un camino menor entre i y k (o k-j) que el encontrado. Entonces modificando ese subcamino se reduciría el peso del camino mínimo Por lo tanto, no era mínimo en primer lugar*

Caminos mínimos finitos

- Si no existen ciclos negativos
  - Entonces el camino mínimo es finito
  - Contiene como mucho |E| ejes (o |V|-1 ejes)

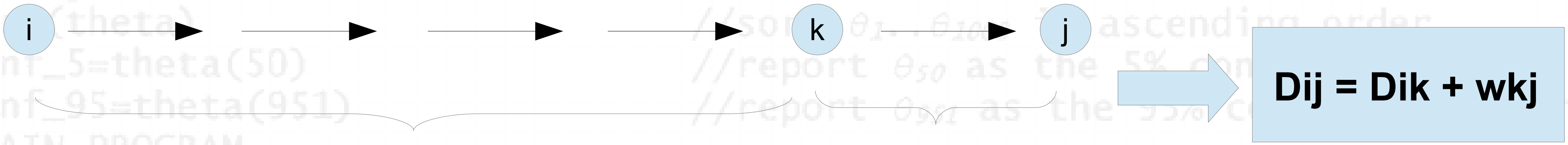


Como el ciclo no es negativo su inclusión aumenta el costo del camino final

Como mucho pasa por los |V| vértices (|V|-1 ejes) Podrían ser todos los ejes disponibles

Descomposición de caminos mínimos

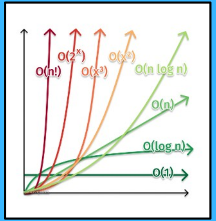
- Sean i,j ∈ V diferentes entre si
  - Cuyo camino mínimo “p” contiene m ejes y un peso D_{ij}
- Si el vértice predecesor a “j” en “p” es “k”
  - Podemos descomponer “p” en dos subcaminos



p’ : es el camino mínimo de i a k con m-1 ejes y peso D_{ik}

Camino mínimo entre k y j y peso w_{kj}

Propuesta de solución



- Sea $L^{(m)}_{ij}$ el peso del camino mínimo del vértice i al j que contiene como mucho m ejes
- Si m=0
  - Únicamente puede llegar al vertice j desde el vertice i con 0 ejes si i=j

$$L_{ij}^{(0)} = \begin{cases} 0 & \text{si } i=j \\ \infty & \text{si } i \neq j \end{cases}$$

- Si m=1

$$L_{ij}^{(1)} = W$$

Propuesta de solución (cont.)

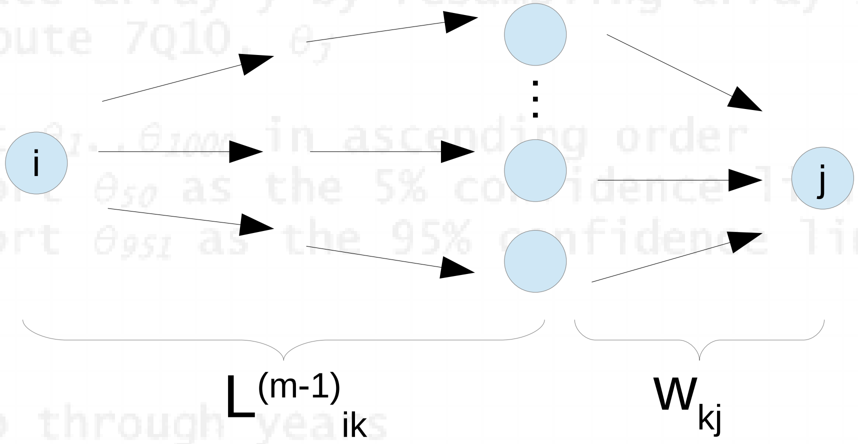
- Si m ≥ 1
  - $L^{(m)}_{ij}$ es el mínimo entre $L^{(m-1)}_{ij}$ y el mínimo de cualquier camino desde i de m-1 ejes a los predecesores de j mas el peso del predecesor a j

$$L_{ij}^{(m)} = min\left(L_{ij}^{(m-1)}, min_{1\le k\le n}\left(L_{ik}^{(m-1)} + w_{kj}\right)\right)$$

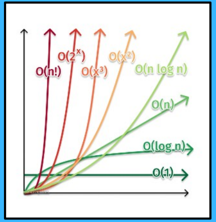
$$L_{ij}^{(m)} = min_{1\le k\le n}\left(L_{ik}^{(m-1)} + w_{kj}\right)$$

Contiene a $L^{(m-1)}_{ij} + w_{jj}$

0



Relación de recurrencia



- En resumen

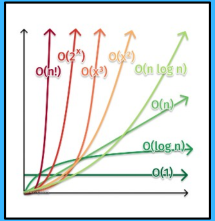
$$L_{ij}^{(0)} = \begin{cases} 0 & \text{si i=j} \\ \infty & \text{si i} \neq \text{j} \end{cases}$$

$$L_{ij}^{(1)} = W$$

$$L_{ij}^{(m)} = min_{1 \leq k \leq n} \left(L_{ik}^{(m-1)} + w_{kj} \right)$$

- Si m=|V|-1 conoceremos el camino mínimo entre todos los vértices

Cálculo $L^{(m)}$ con $L^{(m-1)}$




```
calculo_camino_minimo(L, W)
    Sea n=|V|
    Sea L’ una nueva matriz de nxn

    Desde i=1 a n
        Desde j=1 a n
            L’[i,j]=∞
            Desde k=1 a n
                L’[i,j] = min ( L’[i,j] , L[i,k] + W[k,j] )

    Retornar L’
```

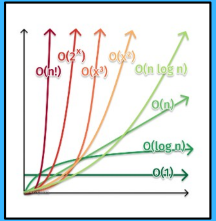
$$L_{ij}^{(m)} = min_{1\le k\le n}\left(L_{ik}^{(m-1)} + w_{kj}\right)$$

Complejidad temporal:

$\Theta(n^3)$

(!) Notar la similitud con la multiplicación de matrices

Solución iterativa



```
ObtenerCaminosMinimos(W)
    Sea n=|V|
    Sea L[1] = W
    Desde m=2 a n-1
        Sea L[m] una nueva matriz de nxn
        L[m] = calculo_camino_minimo(L[m-1], W)

    Retornar L[n-1]
```

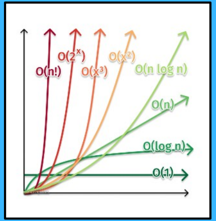
$\Theta(n)$

$\Theta(n^3)$

Complejidad temporal:

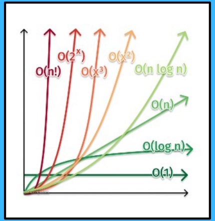
$\Theta(n^4)$

Cálculo de matriz de predecesores



- Partiendo de $L_{ij}^{(m)} = min_{1 \le k \le n} \left(L_{ik}^{(m-1)} + w_{kj} \right)$
  - Podemos ver que para cada i,j terminamos eligiendo aquel vertice k que minimice el peso total.
- Este vértice K corresponde al predecesor de j en el camino mínimo desde i
- Modificando “calculo_camino_minimo(L[m-1], W)”
  - Podemos hacer que guarde en una matriz cual fue el predecedor para cada i,j
- De esa forma utilizando camino_minimo(PI, i, j)
  - Podemos reconstruirlo y mostrarlo

Resumen



- Presentamos un algoritmo
  - que utiliza programación dinámica
- Para calcular el camino mínimo entre todos los vértices
  - Con complejidad temporal $O(|V|^4)$ y complejidad espacial $O(|V|^2)$