DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

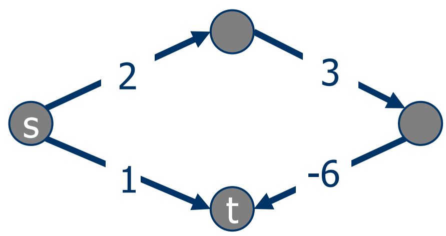
Arbitraje de divisas

Análisis y Diseño de Algoritmos

Algoritmo de Bellman-Ford

Si sólo nos interesan los caminos mínimos desde un vértice concreto del grafo G(V,E), podemos utilizar el algoritmo greedy de Dijkstra, de orden **O(E log V)**, siempre y cuando tengamos **pesos no negativos**.

El algoritmo de Dijkstra no funciona con pesos negativos:



¡Ojo!

Tampoco podemos sumarle una constante a cada peso.

Algoritmo de Bellman-Ford

Si tenemos pesos negativos, podemos utilizar el algoritmo de Bellman-Ford, basado en programación dinámica y de orden **O(EV)**, siempre y cuando no tengamos ciclos de peso negativo:

$$D_i(w) = \begin{cases} \infty & \text{si } i = 0 \\ \min\{D_{i-1}(w), \min\limits_{(v,w)\in E}\{D_{i-1}(v) + c_{vw}\}\} & \text{en otro caso} \end{cases}$$

NOTA: Si un camino de s a t incluye un ciclo de peso negativo, no existe un camino "más corto" de s a t (y, en cualquier caso, Bellman-Ford tampoco encontraría el camino simple más corto).

Algoritmo de Bellman-Ford

Algoritmo de Bellman-Ford: Θ(EV)

```
foreach v ∈ V {
      D[v] = ∞;
      predecesor[v] = null;
}
D[s] = 0;
for (i=1; i<n; i++) {
    foreach (v, w) ∈ E {
        if (D[v] + coste(v,w) < D[w]) {
           D[w] = D[v] + coste(v,w);
           predecesor[w] = v;
        }
    }
}
```

Algoritmo de Bellman-Ford

Algoritmo de Bellman-Ford:

Reconstrucción de los caminos más cortos

```
path (v)
{
  ruta = [v];
  w = v;
  while (predecesor[w]!=null) {
     w = predecesor[w];
     ruta = [w] U ruta;
  }
  return ruta;
}
```


Algoritmo de Bellman-Ford

¿Qué representa $D_i(w)$?

$$D_i(w) = \begin{cases} \infty & \text{si } i = 0 \\ \min\{ D_{i-1}(w) , \min\limits_{(v,\, w)\, \in\, E} \{D_{i-1}(v) + c_{vw}\}\} & \text{en otro caso} \end{cases}$$

- Fijando el origen s, el camino más corto desde s hasta w pasando, como máximo, por i aristas.
- Cuando i=n-1, obtenemos el camino más corto hasta w siempre y cuando no existan ciclos de longitud negativa…

Algoritmo de Bellman-Ford

De forma análoga…

$$D_i(v) = \begin{cases} \infty & \text{si } i = 0 \\ \min\{ D_{i-1}(v)\ ,\ \min\limits_{(v,\,w)\,\in\,E} \{c_{vw} + D_{i-1}(w)\} & \text{en otro caso} \end{cases}$$

- Fijando el destino, el camino más corto desde v hasta t pasando, como máximo, por i aristas.
- Cuando i=n-1, obtenemos el camino más corto desde v siempre y cuando no existan ciclos de longitud negativa…

Ciclos negativos

¿Qué ocurre si existe algún ciclo de longitud negativa?

$$D_i(w) = \begin{cases} \infty & \text{si } i = 0 \\ \min\{ D_{i-1}(w)\ ,\ \min\limits_{(v,\,w)\,\in\,E} \{D_{i-1}(v) + c_{vw}\}\} & \text{en otro caso} \end{cases}$$

$$D_i(v) = \begin{cases} \infty & \text{si } i = 0 \\ \min\{ D_{i-1}(v)\ ,\ \min\limits_{(v,\,w)\,\in\,E} \{c_{vw} + D_{i-1}(w)\} & \text{en otro caso} \end{cases}$$

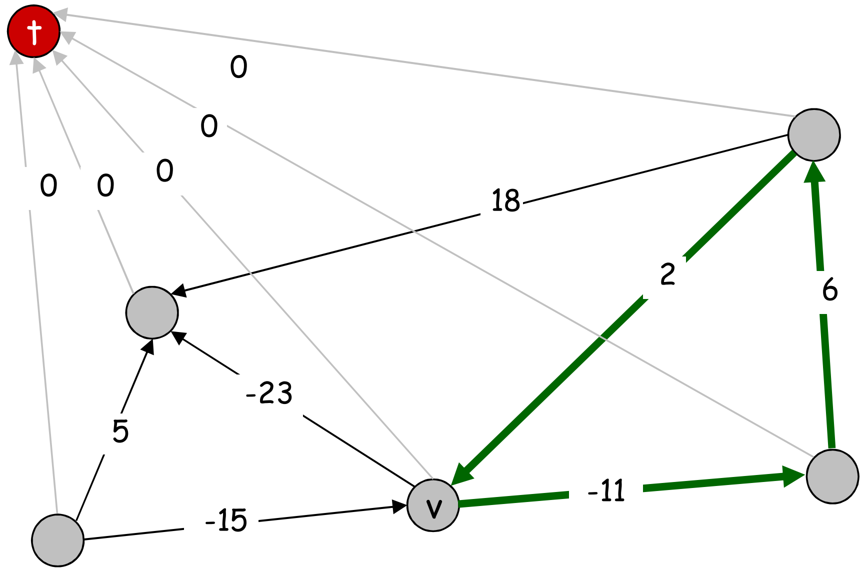
- Al finalizar la ejecución del algoritmo, existirá alguna forma de hacer más corto alguno de los caminos que hemos encontrado, e.g. D(v) + c(v,w) < D(w)

```
foreach (v, w) ∈ E
   if (D[v] + coste(v,w) < D[w])
      print("Existe un ciclo negativo!!!");
```

Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

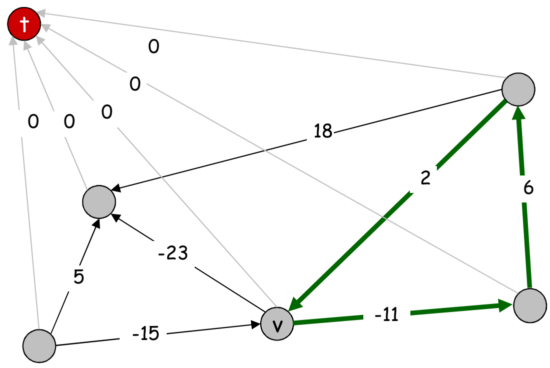
Podemos aumentar el grafo con un nuevo nodo, t, conectado a todos los demás mediante arcos de peso 0:



Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

En el grafo aumentado con t:

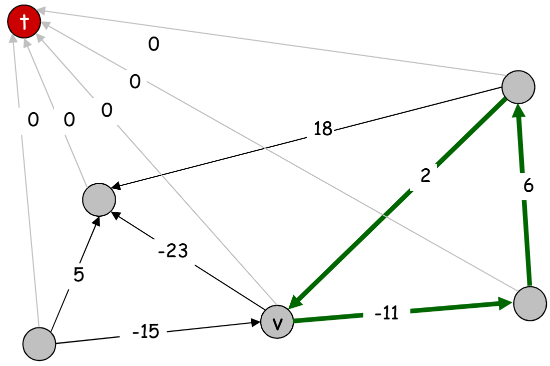


- Si existe un ciclo negativo, el ciclo está conectado con t (todo está conectado con t).
- Dado que ningún arco sale de t, el nodo t no puede formar parte del ciclo negativo; esto es, los ciclos negativos del grafo aumentado son los mismos que los que ya existían en el grafo original.

Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

- Si existen ciclos negativos, podemos hacer los caminos "mínimos" tan cortos como queramos (basta con darle vueltas al ciclo negativo).



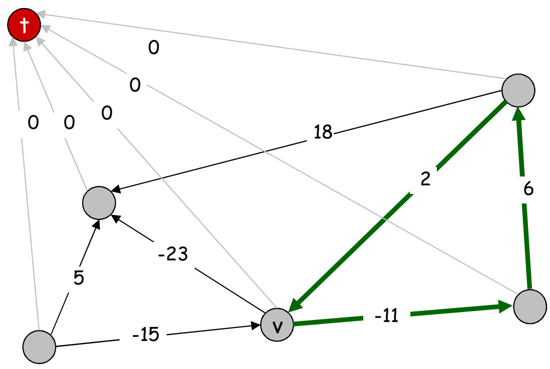
- Si no existen ciclos negativos: $D_n(v) = D_{n-1}(v)$ para todos los nodos v del grafo.

  El algoritmo de Bellman-Ford nos da un método para comprobar si existen ciclos negativos de orden **Θ(EV)**.

Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

Cuando encontramos un nodo v tal que $D_n(v) != D_{n-1}(v)$:



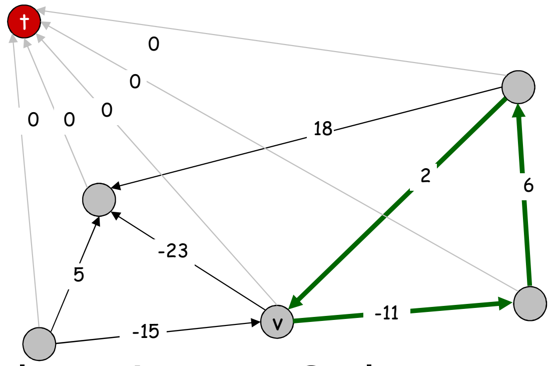
- Para el nodo v, el camino P hasta t incluye n arcos/aristas.

- Dado que un camino simple sólo puede tener n-1 arcos/aristas, el camino "óptimo" P de v a t ha de incluir necesariamente un ciclo, que además es de longitud negativa…

Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

Si tenemos que $D_n(v) \mathrel{!}= D_{n-1}(v)$:

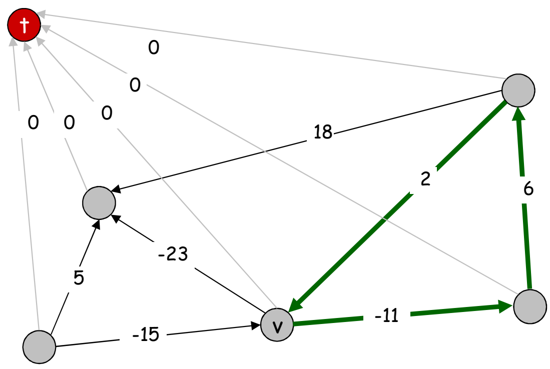


- Todos los caminos de n-1 aristas tienen un coste mayor que el camino P (con ciclo) de n aristas.
- Sea w un nodo que se repite en el camino P y C el ciclo entre las dos ocurrencias de w en P:
  Si el ciclo fuese de longitud positiva, eliminándolo de P obtendríamos un camino más corto con menos de n aristas. Pero, dado que $D_n(v) \mathrel{!}= D_{n-1}(v)$, el ciclo es de longitud negativa.

Ciclos negativos

¿Cómo podemos encontrar un ciclo de longitud negativa?

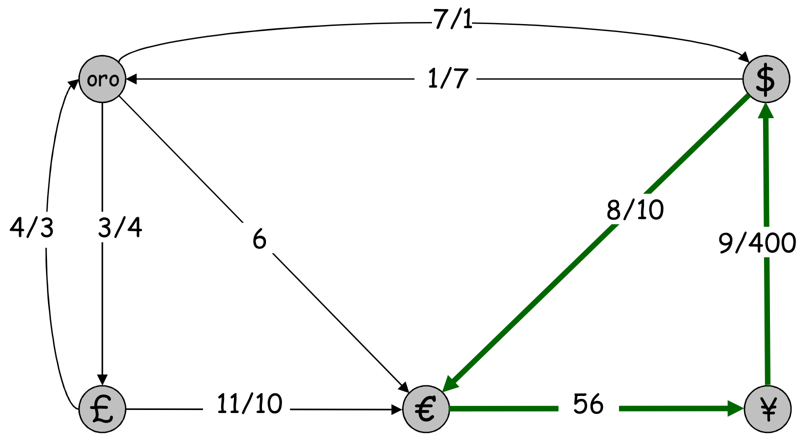
Para reconstruir el ciclo:



- Encontramos un vértice v para el que $D_n(v) \mathrel{!}= D_{n-1}(v)$.
- Encontramos el camino más corto P desde v hasta t.
- Dicho camino P, incluye necesariamente un ciclo C de longitud negativa (entre dos ocurrencias del mismo vértice w en P).

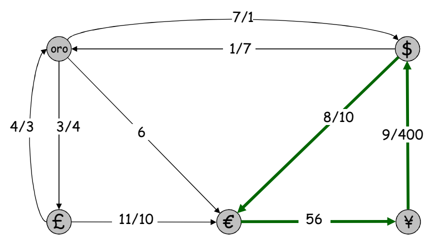
Arbitraje de divisas

La búsqueda de ciclos en grafos
aparece en aplicaciones de trading:



Podemos comprar 5600 yenes con 100 euros,
que luego podemos cambiar por 5600*9/400 = 126 dólares,
que luego podemos convertir de nuevo en 126 * 8/10 = 100.80 euros, obteniendo un beneficio del 0.8% en la operación.

Arbitraje de divisas



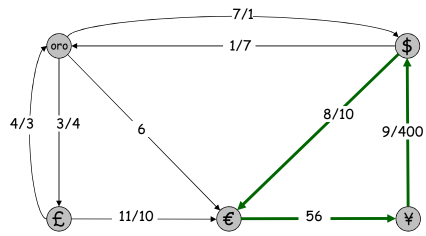
En un grafo en el que aparecen tasas de cambio, el resultado que obtenemos se calcula multiplicando dichas tasas en lugar de sumando pesos.

Un ciclo beneficioso para nosotros será aquél que verifique la siguiente propiedad: $\prod_{(i,j)\in C} r_{ij} > 1$

¿Conocemos alguna operación aritmética que nos permita transformar ese producto de tasas de cambio en una suma?

Arbitraje de divisas

Podemos tomar logaritmos:



Nos interesa: $\prod_{(i,j)\in C} r_{ij} > 1$

… que equivale a: $\sum_{(i,j)\in C} \log r_{ij} > 0$

… y también a: $\sum_{(i,j)\in C} -\log r_{ij} < 0$

Identificar un ciclo beneficioso equivale a encontrar un ciclo de longitud negativa tras aplicar la transformación $\boldsymbol{-\log r_{ij}}$

Arbitraje de divisas

¿Qué podría salir mal?

Los límites del arbitraje incluyen…

- Costes de trading (comprar y vender no es gratis).
- Restricciones legales (permisos, paso de fronteras…).
- Impuestos.
- Costes de transporte y almacenamiento.
- Riesgos de ejecución (alguna de las operaciones falla).
- Riesgo de contraparte (pagar por algo que no se recibe).
- Disrupciones del mercado (p.ej. crisis).
- Costes de capital.