Camino Mínimo

Tecnología Digital V: Diseño de Algoritmos

Universidad Torcuato Di Tella

Agenda del Día

1. Introducción
2. Ejercicio Típico de Camino Mínimo

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con . . .

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con ... BFS.

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con ... BFS.

- Complejidad: $O(n + m)$

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con . . . BFS.

- Complejidad: $O(n + m)$

Problema 2.0

Tenemos un grafo o digrafo (con pesos positivos) en el cual queremos obtener la distancia mínima de un vértice v a cuaquier otro vértice w.

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con . . . BFS.

- Complejidad: $O(n + m)$

Problema 2.0

Tenemos un grafo o digrafo (con pesos positivos) en el cual queremos obtener la distancia mínima de un vértice v a cuaquier otro vértice w.

Este problema lo podemos resolver con . . .

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con . . . BFS.

- Complejidad: $O(n + m)$

Problema 2.0

Tenemos un grafo o digrafo (con pesos positivos) en el cual queremos obtener la distancia mínima de un vértice v a cuaquier otro vértice w.

Este problema lo podemos resolver con . . . Dijkstra.

Problema 1.0

Tenemos un grafo o digrafo (no pesado) en el cual queremos obtener la distancia mínima de un vértice v a cualquier otro vértice w.

Este problema lo podemos resolver con . . . BFS.

- Complejidad: $O(n + m)$

Problema 2.0

Tenemos un grafo o digrafo (con pesos positivos) en el cual queremos obtener la distancia mínima de un vértice v a cuaquier otro vértice w.

Este problema lo podemos resolver con . . . Dijkstra.

- Complejidad: $O(min(m \cdot \log n, n^2))$ [1]

[1] El 30 de julio de 2025 se descubrió un nuevo algoritmo con complejidad de $O(m + nlog^{\frac{2}{3}}(n))$

Algoritmo de Djikstra

1. Asignar distancias tentativas $d_s = 0$ y $d_i = \infty$ para $i \neq s$.
2. Mientras el destino no esté visitado:
    - Seleccionar como nodo actual i el nodo no visitado con menor distancia tentativa, y marcarlo como visitado.
    - Para cada $j \in N^+(i)$ no visitado, calcular $d'_j = d_i + d_{ij}$. Si $d'_j < d_j$ entonces fijar $d_j := d'_j$.
3. Retornar d_t.

Algoritmo de Dijkstra

Dijkstra(G, c, v)
1. Inicializar(G, c, v)

Algoritmo de Dijkstra

DIJKSTRA(G, c, v)

1. INICIALIZAR(G, c, v)
2. $Q = V(G)$

Algoritmo de Dijkstra

DIJKSTRA(G, c, v)

1. INICIALIZAR(G, c, v)
2. $Q = V(G)$
3. **while** $Q \neq \emptyset$ **do**
4. **end while**

Algoritmo de Dijkstra

DIJKSTRA(G, c, v)

1. INICIALIZAR(G, c, v)
2. $Q = V(G)$
3. **while** $Q \neq \emptyset$ **do**
4. $\quad u$ =EXTRAER-MIN(Q)
5. **end while**

DIJKSTRA(G, c, v)

1. INICIALIZAR(G, c, v)
2. $Q = V(G)$
3. **while** $Q \neq \emptyset$ **do**
4. $\quad u$ =EXTRAER-MIN(Q)
5. $\quad$ **for** cada vértice $w \in N(u)$ **do**
6. $\quad$ **end for**
7. **end while**

Algoritmo de Dijkstra

```
DIJKSTRA(G, c, v)
INICIALIZAR(G, c, v)
Q = V(G)
while Q ≠ ∅ do
    u =EXTRAER-MIN(Q)
    for cada vértice w ∈ N(u) do
        RELAJAR(v, w, c)
    end for
end while
```

Fuente del pseudo-código: Introduction to Algorithm, Cormen et al

INICIALIZAR(G, v)

1. $d_v = 0$
2. **for** cada vértice $w \in V(G)$ **do**
3. $\quad d_w = \infty$
4. **end for**

RELAJAR(u, w, c)

1. $d'_w = d_u + c(u, w)$
2. **if** $d'_w < d_w$ **then**
3. $\quad d_w = d'_w$
4. **end if**

Ejercicio 1

Enunciado

Querés viajar desde la ciudad Syrjälä hasta la ciudad Lehmälä en avión. Para eso te dan una red de viajes entre ciudades que se pueden tomar. Cada viaje tiene un costo. Además, los viajes entre ciudades son unidireccionales (es decir, que no tienen viajes de ida y vuelta). Se quieren responder las siguientes preguntas:

a) En caso de haber, declarar una ruta de costo mínimo.

b) En caso de haber, cuales son las diferentes rutas de costo mínimo?

c) Cual es la ruta de costo mínimo con mínima cantidad de vuelos?

d) Cual es la ruta de costo mínimo con máxima cantidad de vuelos?

Pequeña modificación del ejercicio `https://cses.fi/problemset/task/1202`.

Modelamos el problema

Idea: Modelamos el ejercicio como un problema de grafos!

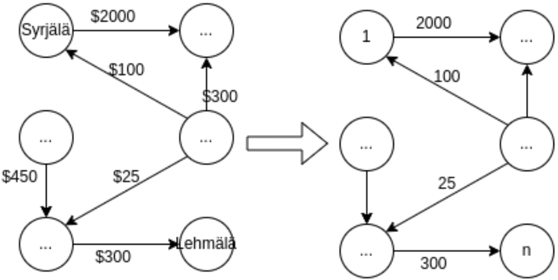
Modelamos el problema

Modelo

Generamos el digrafo pesado $D = (V, E)$ con función de costo w. Donde V es el conjunto de ciudades, estando numeradas del 1 al n siendo 1 la ciudad origen y n la ciudad destino. Luego $v \longrightarrow u \in E$ con costo $w(v \longrightarrow u)$ sii hay un viaje de la ciudad v a la ciudad u y cuesta $w(v \longrightarrow u)$.

Modelo

Generamos el digrafo pesado $D = (V, E)$ con función de costo w. Donde V es el conjunto de ciudades, estando numeradas del 1 al n siendo 1 la ciudad origen y n la ciudad destino. Luego $v \to u \in E$ con costo $w(v \to u)$ sii hay un viaje de la ciudad v a la ciudad u y cuesta $w(v \to u)$.



Ejercicio 1 a

Pregunta: En caso de haber, declarar una ruta de costo mínimo?

Ejercicio 1 a

Pregunta: En caso de haber, declarar una ruta de costo mínimo?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta: En caso de haber, declarar una ruta de costo mínimo?

Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 1.0

Cuál es el camino mínimo desde $v = 1$ hasta $u = n$?

Ejercicio 1 a

Pregunta: En caso de haber, declarar una ruta de costo mínimo?

Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 1.0

Cuál es el camino mínimo desde $v = 1$ hasta $u = n$?

Para responder esto, podremos aplicar Dijkstra?

Ejercicio 1 a

Pregunta: En caso de haber, declarar una ruta de costo mínimo?

Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 1.0

Cuál es el camino mínimo desde $v = 1$ hasta $u = n$?

Para responder esto, podremos aplicar Dijkstra?

Ojo! Se deben cumplir las hipótesis sobre el input del algoritmo:

- Un grafo o un digrafo con aristas de peso positivo.

Ejercicio 1 a

Pregunta: En caso de haber, declarar una ruta de costo mínimo?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 1.0

Cuál es el camino mínimo desde $v = 1$ hasta $u = n$?

Para responder esto, podremos aplicar Dijkstra?
Ojo! Se deben cumplir las hipótesis sobre el input del algoritmo:

- Un grafo o un digrafo con aristas de peso positivo.

Aplicamos Dijkstra sobre v y la respuesta es $d(u)$, siendo d la función de distancia que resulta de aplicar Dijkstra.

Ejercicio 1 b

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo?

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo? Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 2.0

Cuál es el grafo de caminos mínimos desde $v = 1$ hasta $u = n$?

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo? Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 2.0

Cuál es el grafo de caminos mínimos desde $v = 1$ hasta $u = n$?

Cómo encontramos el grafo de caminos minimos?

Ejercicio 1 b

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo? Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 2.0

Cuál es el grafo de caminos mínimos desde $v = 1$ hasta $u = n$?

Cómo encontramos el grafo de caminos minimos?

Pista: Encontrar todas las aristas vu-eficientes. Es decir, todas las aristas que pertenezcan a algún camino mínimo entre u y v. Guia 2, ejercicio 9.

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 2.0

Cuál es el grafo de caminos mínimos desde $v = 1$ hasta $u = n$?

Cómo encontramos el grafo de caminos minimos?
Pista: Encontrar todas las aristas vu-eficientes. Es decir, todas las aristas que pertenezcan a algún camino mínimo entre u y v. Guia 2, ejercicio 9.

9. Dado un digrafo D con pesos $c\colon E(D) \to \mathbb{N}$ y dos vértices s y t, decimos que una arista $v \to w$ es *st-eficiente* cuando $v \to w$ pertenece a algún camino mínimo de s a t. Sea $d(\cdot, \cdot)$ la función que indica el peso de un camino mínimo entre dos vértices. Demostrar que $v \to w$ es st-eficiente si y sólo si $d(s, v) + c(v \to w) + d(w, t) = d(s, t)$.

Pregunta: En caso de haber, cuales son las diferentes rutas de costo mínimo? Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 2.0

Cuál es el grafo de caminos mínimos desde $v = 1$ hasta $u = n$?

Cómo encontramos el grafo de caminos minimos?
Pista: Encontrar todas las aristas vu-eficientes. Es decir, todas las aristas que pertenezcan a algún camino mínimo entre u y v. Guia 2, ejercicio 9.

9. Dado un digrafo D con pesos $c\colon E(D) \to \mathbb{N}$ y dos vértices s y t, decimos que una arista $v \to w$ es *st-eficiente* cuando $v \to w$ pertenece a algún camino mínimo de s a t. Sea $d(\cdot,\cdot)$ la función que indica el peso de un camino mínimo entre dos vértices. Demostrar que $v \to w$ es st-eficiente si y sólo si $d(s,v) + c(v \to w) + d(w,t) = d(s,t)$.

Algún algoritmo para realizar esto?

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \rightarrow w) \in E$.
- En cada iteración, si $t \rightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \rightarrow w$ es uv-eficiente?

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u,t) + c(t \longrightarrow w) + d(w,v) = d(u,v)$.

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \rightarrow w) \in E$.
- En cada iteración, si $t \rightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \rightarrow w$ es uv-eficiente?

Una arista $t \rightarrow w$ es uv-eficiente sii $d(u,t) + c(t \rightarrow w) + d(w,v) = d(u,v)$.

Para conocer esto para cada arista necesitamos.

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u,t) + c(t \longrightarrow w) + d(w,v) = d(u,v)$.

Para conocer esto para cada arista necesitamos.

- $d(u,t)$
- $c(t \longrightarrow w)$
- $d(w,v)$
- $d(u,v)$

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u,t) + c(t \longrightarrow w) + d(w,v) = d(u,v)$.

Para conocer esto para cada arista necesitamos.

- $d(u,t)$ Dijkstra desde u en el grafo original
- $c(t \longrightarrow w)$
- $d(w,v)$
- $d(u,v)$ Dijkstra desde u en el grafo original

Ejercicio 1 b

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u, t) + c(t \longrightarrow w) + d(w, v) = d(u, v)$.

Para conocer esto para cada arista necesitamos.

- $d(u, t)$ Dijkstra desde u en el grafo original
- $c(t \longrightarrow w)$ Función de costo original
- $d(w, v)$
- $d(u, v)$ Dijkstra desde u en el grafo original

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u,t) + c(t \longrightarrow w) + d(w,v) = d(u,v)$.

Para conocer esto para cada arista necesitamos.

- $d(u,t)$ Dijkstra desde u en el grafo original
- $c(t \longrightarrow w)$ Función de costo original
- $d(w,v)$?
- $d(u,v)$ Dijkstra desde u en el grafo original

Algoritmo:
Sea S el conjunto de aristas del grafo de caminos mínimos.

- Recorrer todas las aristas $(t \longrightarrow w) \in E$.
- En cada iteración, si $t \longrightarrow w$ es uv-eficiente, la agregamos a S.
- Luego el grafo de caminos mínimos es $G = (V, S)$. Con V los vértices del grafo original.3

Todo muy lindo, pero cómo determinamos que una arista $t \longrightarrow w$ es uv-eficiente?

Una arista $t \longrightarrow w$ es uv-eficiente sii $d(u, t) + c(t \longrightarrow w) + d(w, v) = d(u, v)$.

Para conocer esto para cada arista necesitamos.

- $d(u, t)$ Dijkstra desde u en el grafo original
- $c(t \longrightarrow w)$ Función de costo original
- $d(w, v)$ Dijkstra desde v en el grafo original traspuesto
- $d(u, v)$ Dijkstra desde u en el grafo original

Ejercicio 1 c

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?

Ejercicio 1 c

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

Pista: Cambiar la función de costo, como?

Ejercicio 1 c

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

Pista: Cambiar la función de costo, como?
Que tiene que tener una función de costo?

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

Pista: Cambiar la función de costo, como?
Que tiene que tener una función de costo?

- Debe tener un dominio sobre todos los nodos
- Su imagen debe poder tener un orden. Es decir, determinar si $c(x) > c(y)$.

Ejercicio 1 c

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

Pista: Cambiar la función de costo, como?
Que tiene que tener una función de costo?

- Debe tener un dominio sobre todos los nodos
- Su imagen debe poder tener un orden. Es decir, determinar si $c(x) > c(y)$.

Propuesta: Usar tuplas de la forma $C(x) = \langle c_x, 1\rangle$. Donde
$\langle a, b\rangle > \langle c, d\rangle \iff a > b \vee (a = c \wedge b > d)$.

Ejercicio 1 c

Pregunta: Cuál es la ruta de costo mínimo con mínima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 3.0

Cuál es el camino mínimo desde u a v con menor cantidad de aristas?

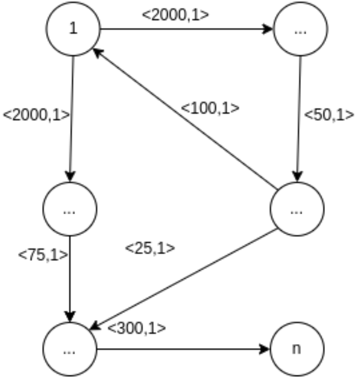
Pista: Cambiar la función de costo, como?
Qué tiene que tener una función de costo?

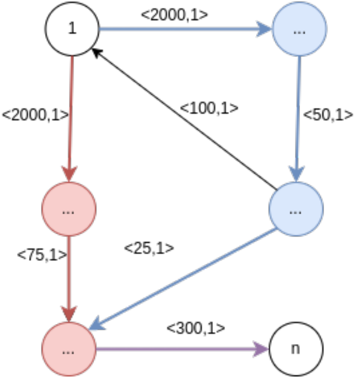
- Debe tener un dominio sobre todos los nodos
- Su imagen debe poder tener un orden. Es decir, determinar si $c(x) > c(y)$.

Propuesta: Usar tuplas de la forma $C(x) = \langle c_x, 1\rangle$. Donde
$\langle a, b\rangle > \langle c, d\rangle \iff a > b \vee (a = c \wedge b > d)$.
Respuesta Realizamos Dijkstra desde u con la nueva función de costo. El camino mínimo es la rama que une a u con v en el árbol que devuelve Dijkstra.

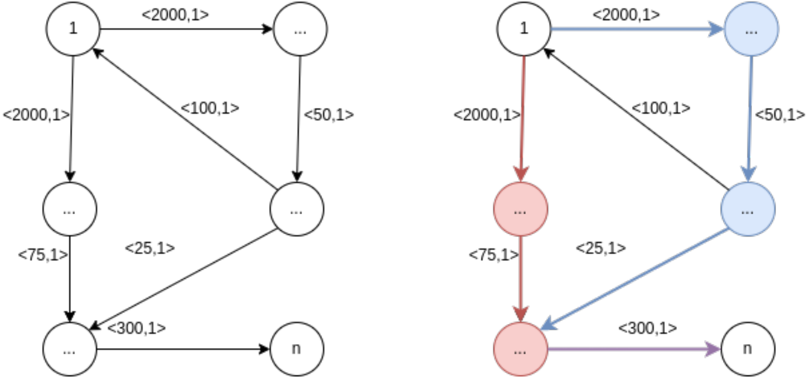
Ejercicio 1 c

Ejemplo



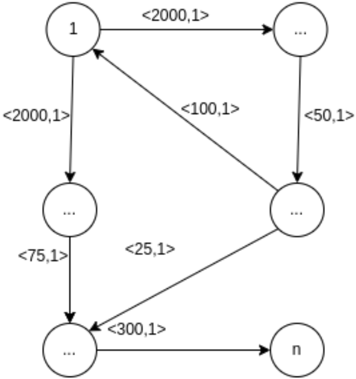


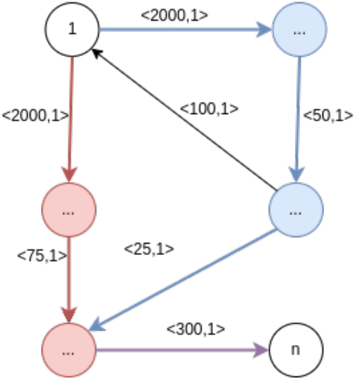
Ejemplo



- $|C1| = \langle 2000, 1\rangle + \langle 75, 1\rangle + \langle 300, 1\rangle = \langle 2375, 3\rangle$.
- $|C2| = \langle 2000, 1\rangle + \langle 50, 1\rangle + \langle 25, 1\rangle + \langle 300, 1\rangle = \langle 2375, 4\rangle$.

Ejemplo





- $|C1| = \langle 2000, 1\rangle + \langle 75, 1\rangle + \langle 300, 1\rangle = \langle 2375, 3\rangle$.
- $|C2| = \langle 2000, 1\rangle + \langle 50, 1\rangle + \langle 25, 1\rangle + \langle 300, 1\rangle = \langle 2375, 4\rangle$.

Luego $|C1| < |C2|$

Ejercicio 1 d

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 4.0

Cuál es el camino mínimo desde u a v con mayor cantidad de aristas?

Ejercicio 1 d

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 4.0

Cuál es el camino mínimo desde u a v con mayor cantidad de aristas?

Pista: Cambiar la función de costo devuelta, cómo?

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 4.0

Cuál es el camino mínimo desde u a v con mayor cantidad de aristas?

Pista: Cambiar la función de costo devuelta, cómo?
La idea es muy parecido a la anterior, solo que quisieramos que a más aristas tenga un camino, menor sea el costo.

Ejercicio 1 d

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 4.0

Cuál es el camino mínimo desde u a v con mayor cantidad de aristas?

Pista: Cambiar la función de costo devuelta, cómo?
La idea es muy parecido a la anterior, solo que quisieramos que a más aristas tenga un camino, menor sea el costo.
Propuesta: Usar tuplas de la forma $C(x) = \langle c_x, -1 \rangle$.

Pregunta: Cuál es la ruta de costo mínimo con máxima cantidad de vuelos?
Responder a esta pregunta es lo mismo que responder en nuestro modelo:

Pregunta 4.0

Cuál es el camino mínimo desde u a v con mayor cantidad de aristas?

Pista: Cambiar la función de costo devuelta, cómo?
La idea es muy parecido a la anterior, solo que quisieramos que a más aristas tenga un camino, menor sea el costo.
Propuesta: Usar tuplas de la forma $C(x) = \langle c_x, -1 \rangle$.
Respuesta Realizamos Dijkstra desde u con la nueva función de costo. El camino mínimo con mayor cantidad de arista es la rama que une a u con v en el árbol que devuelve Dijkstra.