Recorrido Mínimo Uno a Todos

Técnicas de Diseño de Algoritmos

FCEyN UBA

Mayo 2024

¿Dónde estamos?

Hasta ahora vimos:

- Qué son los grafos y cómo representarlos

¿Dónde estamos?

Hasta ahora vimos:

- Qué son los grafos y cómo representarlos
- Algoritmos para **recorrer** y obtener **árbol generador** de un grafo (BFS y DFS)

¿Dónde estamos?

Hasta ahora vimos:

- Qué son los grafos y cómo representarlos
- Algoritmos para **recorrer** y obtener **árbol generador** de un grafo (BFS y DFS)
- Algoritmos para obtener el **árbol generador mínimo** de un grafo (Kruskal y Prim)

¿Dónde estamos?

Hasta ahora vimos:

- Qué son los grafos y cómo representarlos
- Algoritmos para **recorrer** y obtener **árbol generador** de un grafo (BFS y DFS)
- Algoritmos para obtener el **árbol generador mínimo** de un grafo (Kruskal y Prim)

Hoy vamos a ver:

- Algoritmos para obtener el **camino mínimo de uno a todos los nodos** (Dijkstra y Bellman-Ford)

Dijkstra

- Nos da los caminos mínimos de uno a todos en (di)grafos con pesos en las aristas, por lo que también nos da los de uno a uno

Dijkstra

- Nos da los caminos mínimos de uno a todos en (di)grafos con pesos en las aristas, por lo que también nos da los de uno a uno
- Sirve cuando no hay aristas de peso negativo

Dijkstra

- Nos da los caminos mínimos de uno a todos en (di)grafos con pesos en las aristas, por lo que también nos da los de uno a uno
- Sirve cuando no hay aristas de peso negativo
- Su complejidad es $O(min\{m \cdot log(n), n^2\})$

Bellman-Ford

- También nos da los caminos mínimos de uno a todos en (di)grafos pesados.

Bellman-Ford

- También nos da los caminos mínimos de uno a todos en (di)grafos pesados.
- Bellman-Ford detecta los ciclos de peso negativo

Bellman-Ford

- También nos da los caminos mínimos de uno a todos en (di)grafos pesados.
- Bellman-Ford detecta los ciclos de peso negativo
- Su complejidad es $O(m \cdot n)$

Plan de hoy

Recorrido Mínimo
- BFS
- Dijkstra
- Bellman-Ford

Ejercicios
- Policías

Recorrido mínimo

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

BFS

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con BFS?

BFS

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con BFS? **Así nomás, no.**
- ¿Qué tengo que cambiar?

BFS

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con BFS? **Así nomás, no.**
- ¿Qué tengo que cambiar? **El costo tiene que ser el mismo para todas las aristas.**[1]
- ¿Complejidad?

[1] Podemos relajar esto un poco, ver acá.

BFS

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con BFS? **Así nomás, no.**
- ¿Qué tengo que cambiar? **El costo tiene que ser el mismo para todas las aristas.**[1]
- ¿Complejidad? $\boldsymbol{O(|V| + |E|)}$ **sobre lista de adyacencias.**

[1] Podemos relajar esto un poco, ver acá.

Dijkstra

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Dijkstra?

Dijkstra

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c \colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Dijkstra? **Así nomás, no.**
- ¿Qué tengo que cambiar?

Dijkstra

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c \colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Dijkstra? **Así nomás, no.**
- ¿Qué tengo que cambiar? **El costo tiene que ser ≥ 0.**
- ¿Complejidad?

Dijkstra

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Dijkstra? **Así nomás, no.**
- ¿Qué tengo que cambiar? **El costo tiene que ser ≥ 0.**
- ¿Complejidad? $\boldsymbol{O(\min\{|E|\log|V|, |V|^2\})}$ **sobre lista de adyacencias.**

Bellman-Ford

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Bellman-Ford?

Bellman-Ford

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c \colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Bellman-Ford? **Sí!**
- ¿Qué pasa si hay un ciclo con suma de costos negativa?

Bellman-Ford

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Bellman-Ford? **Sí!**
- ¿Qué pasa si hay un ciclo con suma de costos negativa? **Te lo dice, es crack.**
- ¿Complejidad?

Bellman-Ford

Problema

Sea $G = (V, E)$ un (di)grafo con una función de costo para las aristas $c\colon E \to \mathbb{R}$, y v un vértice de G. Para todo $w \in V$, encontrar la mínima suma del costo de las aristas de un recorrido desde v a w.

- ¿Se puede resolver con Bellman-Ford? **Sí!**
- ¿Qué pasa si hay un ciclo con suma de costos negativa? **Te lo dice, es crack.**
- ¿Complejidad? $\boldsymbol{O(|V||E|)}$ **sobre lista de adyacencias.**

Policías

Problema

La nueva reglamentación de una ciudad establece que toda esquina debe estar a lo sumo a 5 cuadras de una estación de policía. Dada la lista de esquinas $\{v_1, \ldots, v_n\}$ de la ciudad, la lista $\{p_1, \ldots, p_k\}$ de esquinas donde hay policías[2], y la lista E de calles, debemos indicar si la normativa se cumple, y en caso contrario cuáles esquinas son las que quedan "desprotegidas".

[2] O sea, asumimos que la policía siempre se ubica en una esquina.

Policías - Resolución

- Primero el modelado: en este caso es bastante directo, las esquinas son los nodos, y las calles los ejes.

Policías - Resolución

- Primero el modelado: en este caso es bastante directo, las esquinas son los nodos, y las calles los ejes.
- Entonces lo que nos pide el enunciado es que todo nodo esté a una distancia menor o igual a 5 de un nodo policía.

Policías - Resolución

- Primero el modelado: en este caso es bastante directo, las esquinas son los nodos, y las calles los ejes.
- Entonces lo que nos pide el enunciado es que todo nodo esté a una distancia menor o igual a 5 de un nodo policía.
- ¿Conocemos algún algoritmo que nos permita resolver este problema?

Policías - Resolución

- Podemos correr un BFS desde cada vértice v y ver la distancia de cada uno a la estación de policía más cercana.

Policías - Resolución

- Podemos correr un BFS desde cada vértice v y ver la distancia de cada uno a la estación de policía más cercana.
- La complejidad de esto es $O(|v| \cdot (m+n))$. ¿Se puede mejorar?

Policías - Resolución

- Podemos correr un BFS desde cada vértice v y ver la distancia de cada uno a la estación de policía más cercana.
- La complejidad de esto es $O(|v| \cdot (m+n))$. ¿Se puede mejorar?
- Si pudiésemos correr un BFS desde todas las estaciones de policia a la vez y cortarlo cuando la distancia es mayor a 5, guardándonos los nodos a los que puedo llegar nos estaríamos guardando todos los vértices que cumplen.

Policías - Resolución

- Podemos correr un BFS desde cada vértice v y ver la distancia de cada uno a la estación de policía más cercana.
- La complejidad de esto es $O(|v| \cdot (m+n))$. ¿Se puede mejorar?
- Si pudiésemos correr un BFS desde todas las estaciones de policia a la vez y cortarlo cuando la distancia es mayor a 5, guardándonos los nodos a los que puedo llegar nos estaríamos guardando todos los vértices que cumplen.
- Pero hay una manera mucho más simple.

Policías - Resolución

- Cambiemos un poquito nuestro modelo: agregamos un **nodo fantasma z** al grafo y lo conectamos con todos los policías.

Policías - Resolución

- Cambiemos un poquito nuestro modelo: agregamos un **nodo fantasma z** al grafo y lo conectamos con todos los policías.
- En este nuevo grafo el problema va a ser un poco distinto. Corriendo BFS desde z y viendo qué nodos están a distancia mayor a 6 nos alcanza para resolverlo (y con complejidad $O(m+n)$!!).

Policías - Resolución

- Cambiemos un poquito nuestro modelo: agregamos un **nodo fantasma z** al grafo y lo conectamos con todos los policías.
- En este nuevo grafo el problema va a ser un poco distinto. Corriendo BFS desde z y viendo qué nodos están a distancia mayor a 6 nos alcanza para resolverlo (y con complejidad $O(m + n)$!!).
- Ahora solo nos queda demostrar que lo que hicimos vale y da la misma respuesta que en el grafo original. Es decir, hay que probar el siguiente lema:

Un nodo v está a distancia menor o igual a 5 de un policía si y solamente si v está a distancia menor o igual a 6 de z

Demostración

En lugar de demostrar ese lema vamos a demostrar algo un poco más fuerte: lo vamos a probar para todo problema de camino mínimo con múltiples orígenes. Es decir, sea z el nodo fantasma y S el conjunto de orígenes queremos ver que
$\forall v \; d(z, v) = min_{s \in S}\{d(s, v)\} + 1$

Demostración

En lugar de demostrar ese lema vamos a demostrar algo un poco más fuerte: lo vamos a probar para todo problema de camino mínimo con múltiples orígenes. Es decir, sea z el nodo fantasma y S el conjunto de orígenes queremos ver que
$\forall v \; d(z, v) = min_{s \in S}\{d(s, v)\} + 1$

- $\forall v \; d(z, v) = min_{s \in S}\{d(z, s) + d(s, v)\}$ ya que z solo está conectado a los nodos S.

Demostración

En lugar de demostrar ese lema vamos a demostrar algo un poco más fuerte: lo vamos a probar para todo problema de camino mínimo con múltiples orígenes. Es decir, sea z el nodo fantasma y S el conjunto de orígenes queremos ver que
$\forall v \; d(z, v) = min_{s \in S}\{d(s, v)\} + 1$

- $\forall v \; d(z, v) = min_{s \in S}\{d(z, s) + d(s, v)\}$ ya que z solo está conectado a los nodos S.
- $\{d(z, s)\} = 1$ ya que z está conectado a todos los orígenes con una arista. Por lo tanto $\forall v \; d(z, v) = min_{s \in S}\{d(s, v) + 1\}$

Demostración

En lugar de demostrar ese lema vamos a demostrar algo un poco más fuerte: lo vamos a probar para todo problema de camino mínimo con múltiples orígenes. Es decir, sea z el nodo fantasma y S el conjunto de orígenes queremos ver que
$\forall v \; d(z, v) = min_{s \in S}\{d(s, v)\} + 1$

- $\forall v \; d(z, v) = min_{s \in S}\{d(z, s) + d(s, v)\}$ ya que z solo está conectado a los nodos S.
- $\{d(z, s)\} = 1$ ya que z está conectado a todos los orígenes con una arista. Por lo tanto $\forall v \; d(z, v) = min_{s \in S}\{d(s, v) + 1\}$
- Si sacamos el 1 hacia afuera queda $\forall v \; d(z, v) = min_{s \in S}\{d(s, v)\} + 1$ tal como queríamos probar.