

DAG's

Igna y Echu

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

2^{do} Cuatrimestre de 2024

Programa de hoy

- 1 Repaso de definiciones y algoritmos
- 2 Ejercicio rayuela

Problemas de CM

Problemas a modelar

- Camino mínimo entre dos vértices
- Camino mínimo entre un vértice y el resto de vértices
- Camino mínimo entre todos los pares de vértices

¿Qué algoritmos resuelven estos problemas?

Problemas de CM

Problemas a modelar

- Camino mínimo entre dos vértices
- Camino mínimo entre un vértice y el resto de vértices
- Camino mínimo entre todos los pares de vértices

¿Qué algoritmos resuelven estos problemas?

Variantes de cada problema

- Grafos con aristas no dirigidas
- Digrafo sin pesos en las aristas
- Digrafo con/sin aristas de peso negativo
- Digrafo con/sin ciclos de peso negativo.

Repaso de propiedades

Principio de optimalidad

Todo sub camino de un camino mínimo es un camino mínimo, siempre que no haya ciclos de pesos negativos.

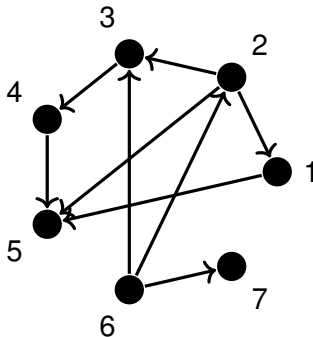
Árbol de caminos mínimos

Sea G un grafo, s un vértice de G y G no posee aristas de costo negativo alcanzables desde s . Sea W el conjunto de vértices alcanzables desde s en G . Entonces existe un árbol orientado T con raíz s y $V(T) = W$, tal que para todo $v \in W$, el camino de s a v en T es un camino mínimo de s a v en G .

DAGs

Definición

Un digrafo D es un DAG (*directed acyclic graph*) si no tiene ciclos dirigidos.

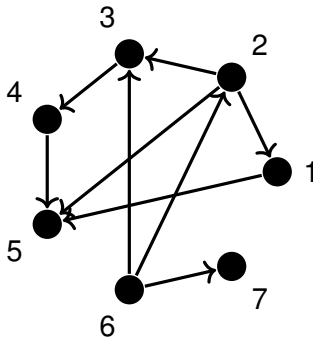


¿Cómo puedo saber si un grafo es un DAG?

DAGs

Definición

Un digrafo D es un DAG (*directed acyclic graph*) si no tiene ciclos dirigidos.



¿Cómo puedo saber si un grafo es un DAG? DFS o BFS, en $O(n + m)$.

DAGs y orden topológico

Orden Topológico

Definimos al orden topológico (o también conocido como *orden de actualización seguro inverso*) de un digrafo $D = (V, E)$ como un ordenamiento de los nodos $v_1 v_2 \dots v_n$ tal que para todo eje $v_i v_j \in E$ vale que $i < j$.

¿Cuál sería un orden topológico para nuestro DAG anterior?

DAGs y orden topológico

Orden Topológico

Definimos al orden topológico (o también conocido como *orden de actualización seguro inverso*) de un digrafo $D = (V, E)$ como un ordenamiento de los nodos $v_1 v_2 \dots v_n$ tal que para todo eje $v_i v_j \in E$ vale que $i < j$.

¿Cuál sería un orden topológico para nuestro DAG anterior? ¿Este es válido $v_6, v_7, v_2, v_3, v_4, v_5, v_1$? ¿Y este $v_6, v_7, v_2, v_3, v_4, v_1, v_5$?

- ¿Todo digrafo tiene un orden topológico?

DAGs y orden topológico

Orden Topológico

Definimos al orden topológico (o también conocido como *orden de actualización seguro inverso*) de un digrafo $D = (V, E)$ como un ordenamiento de los nodos $v_1 v_2 \dots v_n$ tal que para todo eje $v_i v_j \in E$ vale que $i < j$.

¿Cuál sería un orden topológico para nuestro DAG anterior? ¿Este es válido $v_6, v_7, v_2, v_3, v_4, v_5, v_1$? ¿Y este $v_6, v_7, v_2, v_3, v_4, v_1, v_5$?

- ¿Todo digrafo tiene un orden topológico?
- ¿Todo digrafo que admita un orden topológico es un DAG?

DAGs y orden topológico

Orden Topológico

Definimos al orden topológico (o también conocido como *orden de actualización seguro inverso*) de un digrafo $D = (V, E)$ como un ordenamiento de los nodos $v_1 v_2 \dots v_n$ tal que para todo eje $v_i v_j \in E$ vale que $i < j$.

¿Cuál sería un orden topológico para nuestro DAG anterior? ¿Este es válido $v_6, v_7, v_2, v_3, v_4, v_5, v_1$? ¿Y este $v_6, v_7, v_2, v_3, v_4, v_1, v_5$?

- ¿Todo digrafo tiene un orden topológico?
- ¿Todo digrafo que admita un orden topológico es un DAG?
- ¿Todo DAG tiene un orden topológico? ¿Es único ese orden topológico?

DAGs y orden topológico

Orden Topológico

Definimos al orden topológico (o también conocido como *orden de actualización seguro inverso*) de un digrafo $D = (V, E)$ como un ordenamiento de los nodos $v_1 v_2 \dots v_n$ tal que para todo eje $v_i v_j \in E$ vale que $i < j$.

¿Cuál sería un orden topológico para nuestro DAG anterior? ¿Este es válido $v_6, v_7, v_2, v_3, v_4, v_5, v_1$? ¿Y este $v_6, v_7, v_2, v_3, v_4, v_1, v_5$?

- ¿Todo digrafo tiene un orden topológico?
- ¿Todo digrafo que admita un orden topológico es un DAG?
- ¿Todo DAG tiene un orden topológico? ¿Es único ese orden topológico?
- ¿Cuál es la máxima cantidad de ejes que puede tener un DAG de n nodos?

Problemas sobre DAGs

Sobre DAGs muchos problemas son fáciles, gracias al uso del orden topológico. Este lo podemos obtener con el algoritmo de Kahn o invirtiendo el post-order de *DFS*. Por ejemplo:

Problemas sobre DAGs

Sobre DAGs muchos problemas son fáciles, gracias al uso del orden topológico. Este lo podemos obtener con el algoritmo de Kahn o invirtiendo el post-order de *DFS*. Por ejemplo:

Problemas

- ¿Cuál es la cantidad de caminos que van de v a w ?
- Si el digrafo es pesado, ¿Cuál es el camino de menor costo de v a w ?
- ¿Y el de mayor peso?
- ¿Con qué técnica se resuelven estos problemas? ¿Qué propiedad de los caminos mínimos utilizamos?

¿Por qué no podemos hacer lo mismo en digrafos generales?

CM en DAGs

Podemos pensar un algoritmo de camino mínimo para DAGs que sea más eficiente que los que ya conocemos?

CM en DAGs

Podemos pensar un algoritmo de camino mínimo para DAGs que sea más eficiente que los que ya conocemos?

Función recursiva

$$d_v(w) = \begin{cases} 0 & \text{si } w = v \\ \max_{u \in N(w)} (c(w, u) + d_v(u)) & \text{si } cc \end{cases}$$

¿Cuál es la complejidad?

¿Podemos usar alguna técnica de la práctica 1?

CM en DAGs

Podemos pensar un algoritmo de camino mínimo para DAGs que sea más eficiente que los que ya conocemos?

Función recursiva

$$d_v(w) = \begin{cases} 0 & \text{si } w = v \\ \max_{u \in N(w)} (c(w, u) + d_v(u)) & \text{si } cc \end{cases}$$

¿Cuál es la complejidad?

¿Podemos usar alguna técnica de la práctica 1?

¿Por qué no podemos usarla en cualquier grafo?

Problema TAR

Enunciado (Parte 1)

Martín trabaja para la Técnicas de Diseño de Rayuelas co., (TDR), y dentro de poco se organiza el Torneo Argentino de la Rayuela (TAR). Como está muy ocupado nos pidió ayuda para, dada una rayuela, encontrar el casillero inicial desde cuál se puede ganar la mayor cantidad de puntos. Esta rayuela, es de dimensiones $p \times q$, con números distintos escritos en los casilleros. Los números no son consecutivos (es decir, la rayuela puede tener el 1 y el 4 pero no el 2) y tampoco hay un orden claro para recorrerla.

Problema TAR

Enunciado (Parte 2)

El TAR provee las siguientes reglas:

- Estando en una posición (i, j) solo se puede saltar a aquellas posiciones que están en la misma línea horizontal o vertical que (i, j) . Aparte la casilla destino no pueden estar a mayor distancia que k , y su número debe ser mayor al de la casilla (i, j) .
- Al saltar de un casillero con número x a otro con número $x + r$ se ganan r puntos.
- El juego termina cuando le jugadore no puede realizar más saltos válidos.

El input de nuestro problema es la matriz R con los valores de cada casilla de la rayuela y la distancia k .

Modelando ando

- Vamos a tener que crear un digrafo $D = (V, E)$ que nos modele cómo sería un recorrido para esta rayuela. ¿Cómo se les ocurre modelar estos estados?

Modelando ando

- Vamos a tener que crear un digrafo $D = (V, E)$ que nos modele cómo sería un recorrido para esta rayuela. ¿Cómo se les ocurre modelar estos estados?
- Armamos un grafo cuyos vértices $v \in V$ son los pares (i, j) donde $0 \leq i < p$ y $0 \leq j < q$. ¿Que aristas vamos a tener?

Modelando ando

- Vamos a tener que crear un digrafo $D = (V, E)$ que nos modele cómo sería un recorrido para esta rayuela. ¿Cómo se les ocurre modelar estos estados?
- Armamos un grafo cuyos vértices $v \in V$ son los pares (i, j) donde $0 \leq i < p$ y $0 \leq j < q$. ¿Que aristas vamos a tener?
- Conectamos $v = (i_1, j_1)$ con $w = (i_2, j_2)$ con un eje de peso $R[w]$ si se cumple que:
 - v y w están en la misma línea horizontal o vertical y su distancia es menor k : $w = (i_1 + k_1, j_1) \wedge (i_1, j_1 + k_1)$ con $-k \leq k_1 \leq k$.
 - El valor del casillero de w es mayor que el de v : $R[v] < R[w]$.
- ¿Qué representa cada arista? ¿Nos falta representar algún otro estado?

Modelando ando

- Nos falta modelar que se pueda comenzar en cualquier posición y determinar cuándo es que termina. ¿Que se les ocurre?

Modelando ando

- Nos falta modelar que se pueda comenzar en cualquier posición y determinar cuándo es que termina. ¿Que se les ocurre?
- Podemos crear un nodo *inicio* y otro *fin* que representen estos estados. ¿Que aristas van a tener estos?

Modelando ando

- Nos falta modelar que se pueda comenzar en cualquier posición y determinar cuándo es que termina. ¿Que se les ocurre?
- Podemos crear un nodo *inicio* y otro *fin* que representen estos estados. ¿Que aristas van a tener estos?
- *inicio* se va a conectar a todos los vértices de D con costo 0 y todos los vértices $v \in V$ tales que $d_{out}(v) = 0$ (¿Por qué puede pasar esto?) van a tener una arista hacia *fin*.
- Ahora ya tenemos nuestro modelo. ¿Cómo se resuelve este problema? ¿Cumple alguna propiedad nuestro digrafo?

¡Es un DAG!



Solución

- Armamos nuestro digrafo D con sus nodos y aristas pertinentes. Lo cual nos va a costar $\mathcal{O}(p \cdot q \cdot k)$ en el peor caso.

Solución

- Armamos nuestro digrafo D con sus nodos y aristas pertinentes. Lo cual nos va a costar $\mathcal{O}(p \cdot q \cdot k)$ en el peor caso.
- Luego calculamos el camino máximo entre *inicio* y *fin* utilizando el algoritmo de camino máximo en DAG's. $\mathcal{O}(p \cdot q \cdot k)$, pues es lineal en base al tamaño del grafo. ¿Sobre qué nodo vamos a correr nuestra función de camino máximo?
- Por último devolvemos el segundo nodo de algún camino máximo, el cual va a ser un casillero inicial desde el cuál puede obtener la máxima cantidad de puntos. $\mathcal{O}(1)$
- ¿Se puede mejorar el modelo?

Solución

- Armamos nuestro digrafo D con sus nodos y aristas pertinentes. Lo cual nos va a costar $\mathcal{O}(p \cdot q \cdot k)$ en el peor caso.
- Luego calculamos el camino máximo entre *inicio* y *fin* utilizando el algoritmo de camino máximo en DAG's. $\mathcal{O}(p \cdot q \cdot k)$, pues es lineal en base al tamaño del grafo. ¿Sobre qué nodo vamos a correr nuestra función de camino máximo?
- Por último devolvemos el segundo nodo de algún camino máximo, el cual va a ser un casillero inicial desde el cuál puede obtener la máxima cantidad de puntos. $\mathcal{O}(1)$
- ¿Se puede mejorar el modelo? Sí, eso queda para pensar.