PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
Dpto. de Ingeniería de Transporte y Logística

Curso: ICT2233
Semestre: 01'2022
Profesor: Mathias Klapp

INTERROGACIÓN 2
FLUJO EN REDES - ICT2233

TIEMPO: 90 MINUTOS

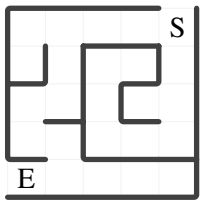
- *La prueba tiene 4 preguntas de diferente puntaje que suman 60 puntos.*
- *¡Coloque nombre en el cuadernillo y no lo separe!* ***Se penalizará con 4 puntos a quien no siga esta instrucción****.*
- *La prueba es sin apuntes y/o equipos electrónicos. Se permite formulario de una hoja escrito a mano.*
- *Dos rondas de pregunta al profesor (al inicio y cuando la prueba lleve un 50% de avance). No hacer preguntas en otro momento por favor para no desconcentrar a sus pares.*

Éxito

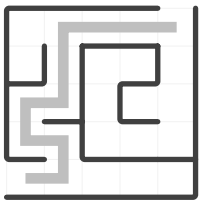
Pregunta 1 (15 puntos)

Un **laberinto** es un juego de ingenio en donde se dibuja una grilla de $n \times n$ cuadraditos sobre una hoja de papel. Todo el perímetro de la grilla es separado del exterior mediante una "muralla" (línea negra gruesa) con excepción de la entrada (E) y salida (S) en la equinas inferior-izquierda y superior-derecha, respectivamente. En el interior también se dibujan murallas horizontales y verticales entre algunos cuadraditos. La Figura (a) dibuja un ejemplo de laberinto de 5×5 cuadraditos.

Una solución del juego consiste en dibujar una línea continua que comience en el centro de E y termine en el centro de S sin tocar murallas. La Figura (b) detalla una solución para nuestro ejemplo.



(a) Laberinto de 5×5



(b) Solución

Diseñe un algoritmo de complejidad polinomial (en función n) que permita a un PC resolver cualquier diseño de laberinto de $n \times n$ cuadraditos que usted reciba. Su algoritmo debe encontrar una solución del laberinto (si existe) o probar que no posee solución. ¿Qué complejidad posee su algoritmo?

PISTA: Modele el problema sobre un grafo apropiadamente definido.

NOTA: No necesita detallar el algoritmo en su respuesta si ya fue visto en clase. Podría simplemente decir "corra el algoritmo X sobre el grafo Y".

Respuesta: Supongamos que cada cuadradito del laberinto es una coordenada (x,y) tal que $x,y \in \{1,\dots n\}$ y que cada muralla interna entre dos cuadraditos (x_1,y_1) e (x_2,y_2) está detallada por un par $\{(x_1,y_1),(x_2,y_2)\}$ en una lista L.

Se puede definir el siguiente grafo dirigido $G := (N,A)$, donde el conjunto N de nodos representa a cada cuadradito

$$N := \{i = (x,y) : x,y \in \{1,\dots n\},$$

y el conjunto de arcos A representa cada movimiento factible de la línea a dibujar (hacia arriba, abajo, izquierda o derecha sin que choque con una muralla interna o externa). Es decir,

$$A := \{((x,y),(x',y')) : (x,y),(x',y') \in N, |x-x'|+|y-y'| = 1, \{(x,y),(x',y')\} \notin L\}$$

Luego, el problema de encontrar una solución equivale a ejecutar búsqueda hacia adelante (BFS o DFS) desde $(1,1)$ y revisar si es que existe una ruta dirigida hacia (n,n). Si la respuesta es no, entonces el problema es infactible, de lo contrario es la solución del laberinto.

La búsqueda en grafos es lineal en el número de arcos. En el peor caso un cuadradito posee 4 movimientos (si no posee murallas), por lo que no hay más de $4 \cdot n^2$ arcos. Es decir, el algoritmo posee complejidad $\mathcal{O}(n^2)$.

Pregunta 2 (8 puntos)

Considere un grafo conexo y dirigido $G = (N,A)$ de n nodos y m arcos, respectivamente. Diseñe o recomiende un algoritmo "rápido" para resolver cada uno de los siguientes problemas asociados a identificar árboles de rutas desde un nodo fuente $s \in N$. Justifique su respuesta.

Nota: Por "rápido" se espera una recomendación de la menor complejidad computacional conocida.

- ◇ (4p) G representa una red vial (nodos son intersecciones, arcos son segmentos de calle) y se busca determinar rutas que minimicen el tiempo total de viaje desde s al resto de los nodos con tiempos $t_a \geq 0$ para cada arco $a \in A$.

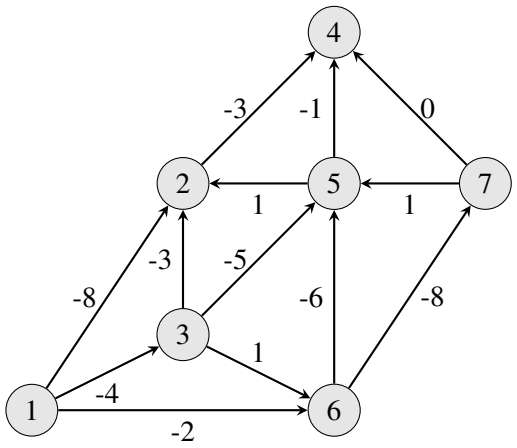
  Respuesta: Dijkstra equipado con un Heap. Posee complejidad $\mathcal{O}(m \cdot \log(n)) \approx \mathcal{O}(m \cdot \log(n))$

- ◇ (4p) G es un grafo genérico y existe un subconjunto de arcos $I \subset A$ indeseables. Se busca determinar rutas desde s al resto de los nodos con la menor cantidad de arcos indeseables.

  Respuesta: Los costos en arcos serían 0 o 1 (indeseado o no), por lo que se recomienda Dijkstra con implementación de Dial en $\mathcal{O}(m + n \cdot C) \approx \mathcal{O}(m)$.

Pregunta 3 (21 puntos)

Considere el siguiente grafo dirigido cuyos costos están detallados sobre cada arco.



Suponga que usted desea encontrar el árbol de rutas de mínimo costo que emanan desde el nodo 1 hacia cada uno de nodos restantes.

(a) (3p) Recomiende el algoritmo más adecuado en cuanto a su complejidad computacional (visto en el curso) que resuelva este problema. ¿Qué complejidad posee?

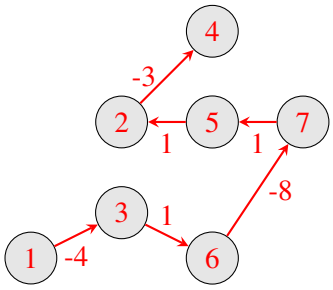
Respuesta: El grafo es acíclico, por lo que se puede ejecutar el algoritmos de Rutas mínimas cerrando nodos en orden topológico. Es $\mathcal{O}(m)$

(b) (8p) Ejecute el algoritmo recomendado por usted (detalle su procedimiento, el vector de distancias y el árbol de rutas obtenido).

Respuesta: Para ejecutar el algoritmo, lo primero es encontrar un orden topológico. Este es: $1-3-6-7-5-6-4$. Luego, se puede ejecutar rutas mínimas cerrando nodos en dicho orden:

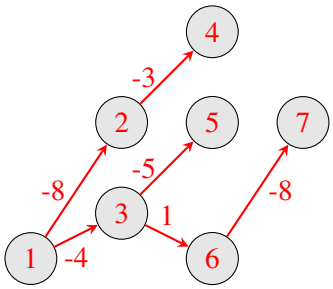
- $d_1^* = 0$
- $d_3^* = d_1^* - 4 = -4$, con $p_3 = 1$
- $d_6^* = \text{mín}(d_1^* - 2; d_3^* + 1) = -3$, con $p_6 = 3$
- $d_7^* = d_6^* - 8 = -11$, con $p_7 = 6$
- $d_5^* = \text{mín}(d_3^* - 5; d_6^* - 6; d_7^* + 1) = -10$, con $p_5 = 7$
- $d_2^* = \text{mín}(d_1^* - 8; d_3^* - 3; d_5^* + 1) = -9$, con $p_2 = 5$
- $d_4^* = \text{mín}(d_2^* - 3; d_5^* - 1; d_7^* + 0) = -12$, con $p_4 = 2$

El árbol de rutas mínimas es el siguiente:



(c) (5p) Ejecute el algoritmo de Dijkstra (sólo detalle el orden de cierre de nodos, vector de distancias y árbol de rutas generado). Compare ambos resultados. ¿Qué problema ve en el algoritmo Dijsktra?

Respuesta: Si se ejecuta Dijkstra, los nodos se cerrarán en este orden: $1, 2, 4, 3, 5, 6, 7$. El vector de distancia sería $[0, -8, -4, -11, -9, -3, -11]$ y el árbol de rutas resultante se detalla en el siguiente árbol:



Cláramente logra una solución subóptima para las rutas desde 1 hacia 2, 4 y 5. LO anterior sucede, pues Dijkstra no garantiza optimalidad en grafos con costos negativos.

(d) (5p) ¿En qué rango de valores debiese estar el costo del arco $(3,2)$ (actualmente en -3) para garantizar que cada árbol de rutas mínimas desde 1 contenga a este arco?

Respuesta: El costo reducido asociado a esta variable es igual a $r_{3,2} = c_{3,2} - (\pi'_2 - \pi'_3) = c_{3,2} - (-9 + 4) = c_{3,2} + 5$. Para que la ctual solución no sea óptima y este arco sea incluido a la base se debe cumplir que $r_{3,2} < 0$, es decir, $c_{3,2} < -5$

Pregunta 4 (16 puntos)

Considere un grafo dirigido (N,A) que posee costos $c_a \geq 0$ para cada arco $a \in A$ y un nodo fuente $s \in N$ desde el cual se desea calcular un árbol de rutas a mínimo costo. Suponga que decide ejecutar el algoritmo de Dijkstra y en este momento ha comenzando una iteración intermedia. El vector actual de distancias es $d \in \mathbb{R}^n$ y la lista de nodos ya cerrados $C \subset N$.

Responda independientemente si cada una de las siguientes afirmaciones es verdadera o falsa. Justifique su respuesta.

(a) (4p) Si j es un nodo abierto tal que $d_j = \text{mín}_{k \in N \setminus C} d_k$, entonces d_j es siempre igual a la distancia de ruta mínima desde s a j.

Respuesta: VERDADERO. Es la definición de un nodo a cerrar que siempre está a min. distancia.

(b) (4p) Si j es un nodo abierto tal que $g_j^- = 1$, entonces d_j es siempre igual a la distancia de ruta mínima desde s a j.

Respuesta: FALSO. Podría suceder que todavía no ha sido alcanzado, es decir, $d_j = \infty$ NOTA DE CORRECCIÓN: Alguien podría decir que es verdadera pues en el enuinciado sale $d \in \mathbb{R}$, es decir, podría decir que se descarta que sea infinito.

(c) (4p) Si j es un nodo abierto tal que $N_j^- \subset C$, entonces d_j es siempre igual a la distancia de ruta mínima desde s a j.

Respuesta: VERDADERO. Verdadero. Todos sus nodos anteriores ya están cerrados por lo que: $d_j = \text{mín}_{(i,j) \in A : i \in C} d_i + c_{i,j}$ es idéntico a $d_j^* = \text{mín}_{(i,j) \in A} d_i + c_{i,j}$

(d) (4p) Si $k \in N$ es el nodo a cerrar por el algoritmo Dijkstra en esta iteración y existe un nodo abierto j tal que $d_k + c_{(k,j)} = d_j$, entonces d_j es siempre igual a la distancia de ruta mínima desde s a j.

Respuesta: FALSO. Se puede crear un contraejemplo evidente (el estudiante debe mostrar un contraejemplo o explicar si desea todo el pontaje).