

DAA

JOSÉ LUIS LEIVA FLEITAS- 412* & EDUARDO GARCÍA MALETA-411¹

CONTENTS

1	Corrupción	2
1.1	Modelacion del problema	2
1.2	Solución 1: Utilizando el algoritmo de Dijkstra	2
1.3	Análisis de Correctitud	3
1.4	Análisis de Complejidad Temporal	3
2	Banco	4
2.1	Paragraphs	4
2.2	Math	4
3	Rompiendo Amistades	5
3.1	Modelación del problema	5
3.2	Figure Composed of Subfigures	8

LIST OF FIGURES

Figure 1	A number of pictures.	9
----------	-----------------------	---

LIST OF TABLES

Table 1	Table of Grades	8
---------	-----------------	---

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

* Department of Biology, University of Examples, London, United Kingdom

¹ Department of Chemistry, University of Examples, London, United Kingdom

1 CORRUPCIÓN

Problema:

Han pasado 20 años desde que Lidier se graduó de Ciencias de la Computación (haciendo una muy buena tesis) y las vueltas de la vida lo llevaron a convertirse en el presidente del Partido Comunista de Cuba. Una de sus muchas responsabilidades consiste en visitar zonas remotas. En esta ocasión debe visitar una ciudad campestre de Pinar del Río.

También han pasado 20 años desde que Marié consiguió su título en MATCOM. Tras años de viaje por las grandes metrópolis del mundo, en algún punto decidió que prefería vivir una vida tranquila, aislada de la urbanización, en una tranquila ciudad de Pinar del Río. Las vueltas de la vida quisieron que precisamente Marié fuera la única universitaria habitando la ciudad que Lidier se dispone a visitar.

Los habitantes de la zona entraron en pánico ante la visita de una figura tan importante y decidieron reparar las calles de la ciudad por las que transitaría Lidier. El problema está en que nadie sabía qué ruta tomaría el presidente y decidieron pedirle ayuda a Marié.

La ciudad tiene n puntos importantes, unidos entre sí por calles cuyos tamaños se conoce. Se sabe que Lidier comenzará en alguno de esos puntos (s) y terminará el viaje en otro (t). Los ciudadanos quieren saber, para cada par s, t , cuántas calles participan en algún camino de distancia mínima entre s y t .

1.1 Modelacion del problema

Representaremos la ciudad como un grafo dirigido $G = V, E$, donde los n puntos importantes de la ciudad estarán representados por n vértices (V) del grafo G y las calles que unen estos puntos estarán representadas por las aristas de dicho grafo (E). Entonces el problema lo analizaremos de la siguiente manera:

Sea $G = V, E$ un grafo, se quiere, para todo par de vértices s y t del grafo, determinar cuántas aristas participan en algún camino de distancia mínima entre s y t .

1.2 Solución 1: Utilizando el algoritmo de Dijkstra

Se le realiza la siguiente modificación al algoritmo de Dijkstra : se le añade un diccionario caminos cuyas llaves son los vértices del grafo y el valor de cada llave será el padre de cada vértice en el camino de distancia mínima entre el vértice de inicio y él. Mediante este diccionario podremos reconstruir luego los caminos de distancia mínima entre el vértice de inicio y los restantes.

Tenemos la función `reconstruir_camino` que se encarga de devolver el camino entre 2 vértices. Esto lo utilizamos en nuestro algoritmo ya sabiendo que el camino entre estos dos vértices es un camino de distancia mínima.

Luego, para darle solución al problema, tenemos la función `solver` que calcula la cardinalidad de algún camino de distancia mínima entre todos los pares de vértices del grafo.

1.3 Análisis de Correctitud

La función `solver` se basa en el algoritmo de Dijkstra para calcular las distancias mínimas desde cada vértice a todos los demás vértices en el grafo. A continuación se desglosan los pasos que se realizan:

- Inicialización :

Se crea un diccionario `longdecaminos` que almacenará la longitud de los caminos mínimos entre cada par de vértices.

- Iteración sobre cada vértice :

Para cada vértice v del grafo, se ejecuta el algoritmo de Dijkstra, obteniendo las distancias y los caminos desde v a todos los demás vértices.

Uso del diccionario de caminos. Este se utiliza para rastrear el camino más corto desde el vértice inicial hasta cada vértice en el grafo. Cada vez que se realiza una operación de relajación en el algoritmo, se actualiza el padre del vértice que está siendo relajado. Esto asegura que, al final del algoritmo, podemos reconstruir el camino más corto desde el vértice inicial hasta cualquier otro vértice utilizando este diccionario.

- Reconstrucción del camino :

Para cada par de vértices v y w , si estos son distintos, se reconstruye el camino mínimo utilizando la función `reconstruircamino`. Luego, si existe un camino, se almacenará la longitud de este en el diccionario

Correctitud :

El algoritmo de Dijkstra garantiza que las distancias calculadas son las mínimas desde el vértice inicial v . Por lo tanto, cualquier camino reconstruido utilizando esas distancias es correcto. Luego, después de reconstruir los caminos con la función `reconstruircamino`, lo que se almacena en el diccionario `longdecaminos` son las longitudes de los caminos de distancia mínima entre todo par de vértices.

1.4 Análisis de Complejidad Temporal

1. Complejidad de Dijkstra:

La complejidad temporal del algoritmo Dijkstra utilizando una cola de prioridad (heap) es $O((V + E)\log(V))$, donde V es el número de vértices y E es el número de aristas en el grafo.

2. Llamadas a Dijkstra:

En nuestra función `Solver`, se llama a Dijkstra desde cada vértice, lo que implica que se ejecuta V veces. Por lo tanto, la complejidad total por esta parte es $O(V * (V + E)\log(V))$.

3. Reconstrucción del camino:

La reconstrucción del camino tiene una complejidad lineal en función del número de vértices en el camino, que en el peor de los casos es $O(V)$. Sin embargo, esta función se llama para cada par de vértices, lo que significa que se invoca una vez por cada combinación de vértices diferentes (v, w) . Por tanto, son un total de $V * (V - 1)$ invocaciones. Esto implica que la complejidad total derivada de la reconstrucción es $O(V^3)$.

Sumando todo, la complejidad total del algoritmo `solver` es: $O(V^2 * \log(V) + V * E * \log(V)) + O(V^3)$.

En un grafo denso donde E puede ser aproximadamente V^2 , el caso peor sería $O(V^3)$

A statement requiring citation [?].

2 BANCO

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

1. First item in a list
2. Second item in a list
3. Third item in a list

2.1 Paragraphs

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

PARAGRAPH DESCRIPTION Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

DIFFERENT PARAGRAPH DESCRIPTION Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

2.2 Math

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc

elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

$$\cos^3 \theta = \frac{1}{4} \cos \theta + \frac{3}{4} \cos 3\theta \quad (1)$$

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Definition 1 (Gauss). To a mathematician it is obvious that $\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$.

Theorem 1 (Pythagoras). *The square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides.*

Proof. We have that $\log(1)^2 = 2\log(1)$. But we also have that $\log(-1)^2 = \log(1) = 0$. Then $2\log(-1) = 0$, from which the proof. \square

3 ROMPIENDO AMISTADES

Por algún motivo, a José no le gustaba la paz y le irritaba que sus compañeros de aula se llevaran tan bien. Él quería ver el mundo arder. Un día un demonio se le acercó y le propuso un trato: "A cambio de un cachito de tu alma, te voy a dar el poder para romper relaciones de amistad de tus compañeros, con la única condición de que no puedes romperlas todas". Sin pensarlo mucho (Qué más da un pequeño trocito de alma), José aceptó y se puso a trabajar. Él conocía, dados sus k compañeros de aula, quiénes eran mutuamente amigos.

Como no podía eliminar todas las relaciones de amistad, pensó en qué era lo siguiente que podía hacer más daño. Si una persona quedaba con uno o dos amigos, podría hacer proyectos en parejas o tríos (casi todos los de la carrera son así), pero si tenía exactamente tres amigos, cuando llegara un proyecto de tres personas, uno de sus amigos debería quedar afuera y se formaría el caos.

Ayude a José a saber si puede sembrar la discordia en su aula eliminando relaciones de amistad entre sus compañeros de forma que todos queden, o bien sin amigos, o con exactamente tres amigos.

3.1 Modelación del problema

Representaremos los k compañeros de José como los vértices de un grafo no dirigido $G = V, E$ de tamaño k , donde las relaciones de amistad entre ellos están representadas por las aristas entre los vértices del grafo. Entonces analizaremos el problema de la siguiente forma :

Es posible eliminar aristas del grafo G (sin eliminarlas todas) de manera tal que todos los vértices queden, o bien aislados, o con exactamente 3 vecinos.

También podemos verlo de la siguiente manera:

Dado un grafo G determinar si existe un subgrafo G' de G tal que G' es regular de grado 3.

3.1.1 *Np-Compleitud*

Para demostrar que un problema es NP – Completo hay que demostrar que:

1. El problema está en NP.
2. Se puede reducir un problema conocido como NP – Completo a este problema en tiempo polinómico.

Paso1: Probar que el problema de encontrar un subgrafo regular de grado 3 (SR₃) está NP

Dado un grafo G y un conjunto de aristas que supuestamente forman un subgrafo regular de grado 3, podemos verificar en tiempo polinomial si este subgrafo es regular de grado 3. Para ello, recorremos todas las aristas del subgrafo y contamos el grado de cada vértice. Si todos los vértices tienen grado 3, entonces el subgrafo es regular de grado 3. Este procedimiento tiene un costo de $O(|V| + |E|)$, siendo $|V|$ la cantidad de vértices y $|E|$ la cantidad de aristas.

Paso2: Demostrar que SR₃ es NP-Completo

Para demostrar que el problema CSP es NP-completo, se necesita realizar una reducción desde un problema conocido como NP-completo. En este caso, se utilizará el problema de 3-coloración.

Problema de 3-coloración

Dado un grafo no dirigido $G = (V, E)$ determinar si existe una partición del conjunto de vértices V , de la forma V_1, V_2, V_3 tal que no existen aristas entre los vértices que pertenecen a la misma partición V_i .

Definiciones:

- G : Grafo de entrada del problema de 3-coloración.
- $V(G)$: Conjunto de vértices del grafo G .
- $E(G)$: Conjunto de aristas del grafo G .
- $d(v_i)$: Grado del vértice i .
- K'_n : Grafo completo de n vértices al que se le quita una arista (este grafo tiene todos los vértices con grado $n-1$, excepto dos vértices que tienen grado $n-2$).
- 3-partición : Dividir el conjunto de vértices en 3 conjuntos disjuntos.

Paso3: Transformación de la entrada

A partir del grafo G , se construye un nuevo grafo G' siguiendo los pasos siguientes:

1. **Ciclos** : Por cada vértice v_i en $V(G)$, se crean 3 ciclos (denotados como : C_i^1, C_i^2, C_i^3), cada uno con una longitud de $2 * d(v_i) + 1$. Los vértices de cada ciclo son denotados como c_i^h , $1 \leq i \leq n, 1 \leq d \leq 2 * d(v_i) + 1, 1 \leq h \leq 3$.
2. **Subgrafos** : Por cada arista $e_j \in E(G)$ se crean 3 subgrafos (D_j^1, D_j^2, D_j^3), donde cada uno es un K'_4 . Los dos vértices con grado 2 en cada subgrafo son denotados como x_i^h y y_j^h .
3. **Conexiones** : Por cada arista entre los vértices v_s y v_t en G :

- Se seleccionan dos vértices $c_s^h a$ y $c_s^h b$ ($c_s^h a$ y $c_s^h b$) que aún tengan grado 2 de los ciclos correspondientes a v_s y v_t , C_s^h (C_t^h).
- Por cada $1 \leq h \leq 3$, se agregan a G' las aristas $\langle c_s^h a, x_j^h \rangle$, $\langle c_t^h a, y_j^h \rangle$, $\langle c_t^h a, x_{hj} \rangle$ y $\langle c_t^h b, y_j^h \rangle$.

Una vez consideradas todas las aristas en el paso (3), se tiene que para cada ciclo C_i^h ($1 \leq i \leq n, 1 \leq h \leq 3$) solo queda un vértice con degree 2. Nombremos dichos vértices como w_i^h .

4. Construcción Adicional: Por cada $1 \leq i \leq n$, se construye un subgrafo U_i , que es un K_4' más un vértice al que denominaremos u_i , los vértices de grado 2 del K_4' los denominaremos x_i y y_i , el vértice u_i se une a los restantes del K_4' mediante una arista $\langle x_i, u_i \rangle$.

Se toman todos los grafos U_i y se agregan a G' . junto con las aristas $\langle u_i, w_i^1 \rangle$, $\langle y_i, w_i^2 \rangle$ y $\langle y_i, w_i^3 \rangle$.

5. Ciclo final : Se agrega el ciclo C' de longitud $2n$ a G' , conformado por los vértices $a_1, 1, \dots, a_1, n, a_2, 1, \dots, a_2, n$ y se agregan las aristas $\langle a_p, u_i \rangle$ para $p = 1, 2$.

Paso4: Demostrar que G es 3-coloreable si y solo si G' tiene un subgrafo regular de grado 3

(\Rightarrow) Sea G un grafo 3-coloreable, y una 3-partición de $V(G)$ tal que en cada conjunto V_i queden solo vértices de un mismo color, entonces existe un grafo H , subgrafo inducido de G' que es 3-regular, ya que siempre podemos tomar los vértices de la siguiente forma:

1. Todos los vértices a_i, j están en H

3.1.2 Solución