

Trabajo Práctico Anual

de Matemática Discreta

Sección 3: Grafos

**Alternativa elegida: Alternativa para programadores**

**Ejercicio Nº1:** Realizar un algoritmo de ‘camino más corto’ (puede ser con código ya creado por otra persona) que resuelva problema de este tipo. La modelización puede ser de nodos que contengan caminos que salen de él y los costos de cada camino. Si se quiere agregar heurísticas, mejor.

**Resolución:**

Los grafos son conjuntos conformados por unidades llamadas vértices o nodos, los cuales se relacionan entre sí o unen a través de aristas o arcos. Los grafos se representan tradicionalmente como puntos unidos por líneas, que determinan los vértices y las aristas respectivamente.

Las relaciones de los vértices, es decir sus aristas, a veces están determinadas con un peso, un valor numérico que determina la relación entre los vértices que unen, denominados vértices adyacentes. En conjunto, estas relaciones determinan caminos, que unen varios vértices entre sí. Con respecto a los pesos mencionados anteriormente, estos caminos tendrán un valor o peso asociado, el cual resulta de sumar los pesos de las aristas que conforman el camino.

Dicho esto, es posible explicar el problema y su resolución. Lo que se nos plantea es que dado un grafo, con cierta cantidad de vértices y con dichos vértices unidos por aristas, las cuales tienen un peso asociado a cada una de ellas. Si se buscase encontrar un camino que una 2 vértices, es posible llegar a la conclusión de que hay varios caminos que posibilitan la unión de dichos vértices. Sin embargo, lo que el ejercicio demanda es un algoritmo que posibilite determinar cuál camino es el más corto, es decir, aquel cuya longitud (suma del peso de las aristas) sea menor a la de los demás caminos.

Esta problemática en el estudio de los grafos ha dado como resultado que numerosos autores produjeran sus propios algoritmos para poder resolver este problema.

Los algoritmos más importantes para resolver este problema son:

* [**Algoritmo de Dijkstra**](http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra)**, resuelve el problema de los caminos más cortos desde un único vértice origen hasta todos los otros vértices del grafo.**
* [Algoritmo de Bellman - Ford](http://es.wikipedia.org/wiki/Algoritmo_de_Bellman-Ford), resuelve el problema de los caminos más cortos desde un origen si la ponderación de las aristas es negativa.
* [Algoritmo de Búsqueda A\*](http://es.wikipedia.org/wiki/Algoritmo_de_búsqueda_A*), resuelve el problema de los caminos más cortos entre un par de vértices usando la [heurística](http://es.wikipedia.org/wiki/Heurística) para intentar agilizar la búsqueda.
* [Algoritmo de Floyd - Warshall](http://es.wikipedia.org/wiki/Algoritmo_de_Floyd-Warshall), resuelve el problema de los caminos más cortos entre todos los vértices.
* [Algoritmo de Johnson](http://es.wikipedia.org/wiki/Algoritmo_de_Johnson), resuelve el problema de los caminos más cortos entre todos los vértices y puede ser más rápido que el de Floyd - Warshall en grafos de baja densidad.
* [Teoría perturbacional](http://es.wikipedia.org/wiki/Teoría_perturbacional), encuentra en el peor de los casos el camino más corto a nivel local

Para resolver el ejercicio del trabajo práctico utilizamos el primer algoritmo utilizado, el algoritmo de Dijkstra, desarrollado por el  físico holandés [Edsger Dijkstra](http://es.wikipedia.org/wiki/Edsger_Dijkstra). El funcionamiento de dicho algoritmo está bien ilustrado en el documento de anexo “Algoritmo de Dijkstra.docx”.

El algoritmo en sencillas palabras evalúa cada posible camino entre el vértice inicial, denominado vértice origen, y un vértice final o vértice destino, compara estos caminos entre si y determina cuál de ellos tiene la menor distancia. Sobre la base de este algoritmo desarrollamos un programa capaz de resolver el problema.

**Manual de uso del programa:**

El programa solicita al usuario que ingrese primero la cantidad de vértices que hay en el grafo y la cantidad de aristas que haya. El usuario debe ingresar la cantidad de vértices, dejar un espacio (presionar la tecla de espacio) y luego poner la cantidad de aristas, finalmente presionar la tecla de *enter*.

Posteriormente se debe ingresar los vértices y las aristas. Los vértices quedan determinados por la cantidad ingresada anteriormente, es decir, si por ejemplo la cantidad ingresada fue 5, los vértices del grafo serán 1-2-3-4-5. Así quedan denominados los vértices, mientras que las aristas quedaran determinadas por los pesos o valores de las mismas.

El modo de introducción de los vértices y aristas consta de tres datos. Primero, se ingresa el vértice de origen, se deja un espacio, se ingresa el vértice adyacente al ingresado anteriormente, se vuelve a dejar otro espacio y finalmente se ingresa el peso de la arista que une a los vértices ingresados. Al presionar la tecla *enter*, el programa solicitará que ingrese un vértice inicial, el cual será el primer vértice del camino a buscar. Luego de ser ingresado y de que se presione la tecla *enter*, el programa mostrará por pantalla la distancia más corta que hay entre el vértice inicial y cada uno de los demás vértices del grafo.

Siguiente a ello, el programa solicitará al usuario que ingrese el vértice final, que es el vértice final del camino buscado. Entonces el camino quedará determinado por el vértice inicial y final, y todos los vértices por los que pasa el camino para llegar de uno al otro. Luego de que se ingresa el vértice final y se presiona la tecla *enter*, el programa mostrara por pantalla el camino más corto entre el vértice inicial y el final, y los vértices por los que pasa dicho camino.

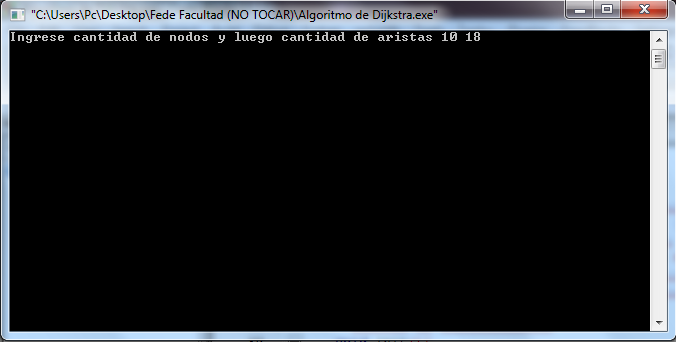
Ahora es necesario aclarar ciertas cosas con respecto al programa:

Aclaración 1: Este programa funciona siempre y cuando haya un camino posible entre cualquier vértice con cualquiera de los demás. Es decir que no están permitidos los vértices aislados. El uso de grafos que no cumplan con esto hará que el programa arroje valores inciertos.

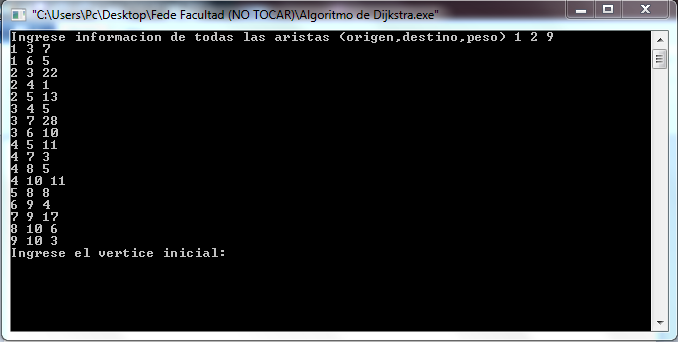
Aclaración 2: En el momento en que se ingresan los vértices con sus aristas correspondientes, el orden en que se ingresan los vértices es independiente del resultado. Es decir que el usuario podría ingresar la combinación 1 2 8 (es decir, vértice 1, el vértice 2 y la arista de peso 8) o bien 2 1 8, y no afectará el resultado.

**Manual de uso paso por paso:**

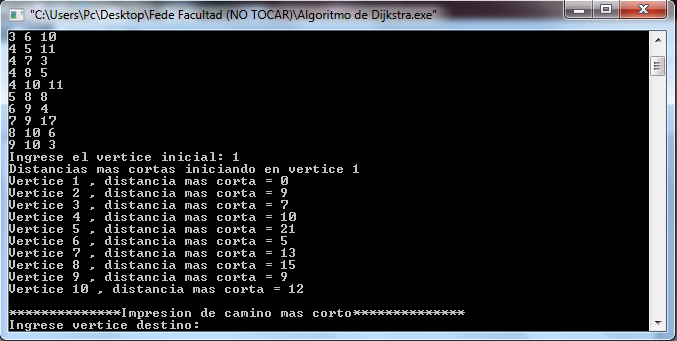
1. **Se ingresa la cantidad de vértices y aristas del grafo y se presiona *enter*.**



1. **Se ingresan los vértices y las aristas correspondientes, de la manera mencionada (un vértice, espacio, otro vértice, espacio y la distancia de la arista que los une), luego se presiona la tecla *enter*.**



1. **Se ingresa el vértice inicial del camino y se presiona *enter*, luego de esto el programa mostrará por pantalla la distancia más corta hacia los demás vértices del grafo.**



1. **Se ingresa el vértice final del camino y se presiona *enter*, lo que hará que el programa muestre por pantalla el camino más corto del vértice inicial al vértice final.**

