# ALL ROAD LEAD WHERE

Sara Chica, Rodrigo Gualtero

15 de Octubre, 2012

## Índice

**1. Introducción** **1**

**2. Definición del problema** **2**
2.1. Entrada . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2.2. Salida . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

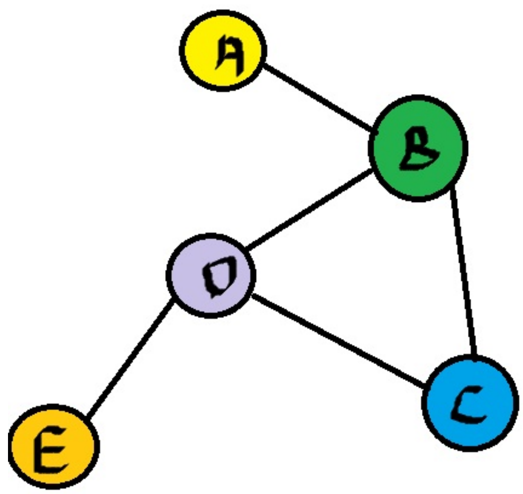
**3. Modelamiento matemático** **3**

**4. Planteamiento de la Solución** **3**

**5. Conclusiones** **4**

## 1. Introducción

Este es un problema de la UVA, identificado con el código *10009*, El cual consiste en encontrar el camino más corto entre dos ciudades evitando atravesar muchas ciudades en el recorrido.
Cada ciudad se encuentra representada por una letra mayúscula.
A continuación se presenta un ejemplo en donde se puede observar más claramente lo dicho anteriormente:




La imágen representa el mapa de 5 ciudades si deseo ir de la ciudad $A$ a la ciudad $E$ el camino más corto debe ser de $A$ *hasta* $E$ pasando por $B$ *y* $D$.

# 2. Definición del problema

Para este problema el mapa de las ciudades se puede modelar como un grafo donde se muestran las conexiones entre las ciudades. Este grafo cumple varias caracteristicas:

1. Un nodo no puede conectarse consigo mismo.
2. El grafo es no dirigido, es decir siendo $A$ y $B$ ciudades si $A$ está conectada con $B$ entonces $B$ estará conectada con $A$.
3. El grafo no siempre es conexo, porque no necesariamente podemos encontrar un camino de $A$ hasta la ciudad $B$.

## 2.1. Entrada

En un principio recibimos un entero que representa el número de (mapas) que serán ingresados. Seguido a esto entran dos enteros $X$ y $Y$ donde $X$ representa el número de parejas de ciudades que están conectadas y $Y$ el número de consultas entre pares de ciudades siendo la primera ciudad destino y la siguiente origen. Por último se reciben dos cadenas separadas por un espacio cada una con el nombre de la ciudad respectivamente.

## 2.2. Salida

Se debe imprimir una cadena que muestre todas las ciudades que tuvo que atravesar el viajero para llegar de una ciudad $A$ a una cuidad $B$ incluyendo la ciudad $A$ $y$ $B$; cada una de estas ciudades se identifica con su letra inicial en mayúscula.

Ejemplo:
$TPQRS$
$BMTE$
$FRTP$

# 3. Modelamiento matemático

Sea G un grafo, $G(V, L)$, donde

V es el conjunto de nodos del grafo,

$$V = \{a_1, a_2, ..., a_n\}$$

y L es el conjunto de todas las conexiones,

$$L = \{(a_1, a_2), (a_2, a_3), ..., (a_{n-1}, a_n)\}$$

El grafo es no dirigido si $(a_i, a_{i+1}) = (a_{i+1}, (a_i)$.
No existe una conexión tal que $(a, a)$; es decir $(a, a) \notin L$

# 4. Planteamiento de la Solución

Para determinar la solución del problema se debe emplear un algoritmo que se encuentra implementado sobre grafos comunmente conocido como BFS o (Búsqueda por anchura) el cual consiste en encontrar todos los caminos desde un nodo origen hasta cada uno de los nodos existentes en el grafo; una vez se tienen estos caminos en un vector llamado camino solo basta con realizar una función recursiva a la cual se le envíe el nodo origen junto con el nodo destino y mientras que el nodo origen sea diferente del nodo destino se irá imprimiendo el retorno de camino hasta cuando sean iguales origen y destino.

# 5. Conclusiones

1. Es importante investigar sobre este tipo de algoritmos de recorridos sobre grafos porque reducen el tiempo de ejcución y la complejidad de las búsquedas sobre los mismos.

2. Estos algoritmos que resuelven problemas en donde se debe encontrar el camino más corto, tienen aplicaciones en redes y telecomunicaciones, facilitan el diseño de sistemas, entre otros.

3. Este tipo de problemas permite que los estudiantes se den cuenta de la eficiencia que puede tener la representación de información en un grafo si esta requiere búsquedas complejas y soluciones eficientes; otro aspecto que se debe tener encuenta al momento de utilizar un grafo como contenedor, es la cantidad de información que se debe almacenar ya que si es mucha este tipo de implementación no será la más adecuada.