

FECHA LÍMITE DE ENTREGA: 13 DE DICIEMBRE

Números de Bacon

Queremos tener una funcionalidad similar a la de la página web <http://oracleofbacon.org>, donde se intenta conectar a un actor con Kevin Bacon (o cualquier otro que actúe como origen) en el menor número de saltos, donde un salto entre dos actores puede darse si han participado juntos en alguna película.



Primer paso: Para ello primero implementaremos una clase para representar grafos cuyos vértices están identificados mediante strings.

```
class GrafoNombres {
private:
    Tabla<string, uint> tn; // string -> index
    string* nombres;      // index -> string
    Grafo* _G;            // grafo con vértices numerados

public:
    GrafoNombres(string filename, string delimiter);
    ~GrafoNombres();
    bool contiene(string s) const; // existe un vértice de nombre s?
    int indice(string s) const;    // devuelve el número asociado al vértice s
    const string& nombre(int v) const; // devuelve el nombre asociado al número v
    const Grafo& G() const;        // devuelve el grafo de números
};
```

Esta clase tiene como atributos una tabla (la implementada en EDA) que asocia números (los valores) a cadenas (las claves), un array para realizar la asociación inversa, y un grafo cuyos vértices son números (los vistos en clase). El constructor recibe como argumentos el nombre de un fichero y

un delimitador de nombres y construye el grafo. El formato del fichero es el siguiente: cada línea contiene una serie de nombres separados por el delimitador y representa aristas entre el primer nombre y todos los demás en la línea (véase el fichero `movies.txt` en el campus virtual, donde el primer nombre de cada línea corresponde a una película y el resto a sus intérpretes). El resto de métodos son de consulta y permiten trabajar con el grafo con la dualidad nombre-número.

Segundo paso: A continuación, utilizando la clase anterior y las vistas en clase, implementaremos dos funciones para utilizar en el programa principal. La primera es la que permite encontrar un camino mínimo entre un actor origen (recibido como argumento por la función) y una serie de actores destino (introducidos por teclado).

```
void GradoSeparacion(const string& origen) {  
    GrafoNombres pelis("movies.txt", "/");  
    ...  
}
```

Esta función primero construye un grafo de nombres a partir del fichero `movies.txt` (con casi 4 200 películas y más de 115 000 actores) y posiblemente algún otro objeto. A continuación entra en un bucle en el que se solicita al usuario un nombre de actor y, si el actor existe y está conectado al origen, se muestra su grado de separación (su número de Bacon) y el camino mínimo que los une. El bucle termina cuando el nombre de actor introducido es vacío. Por ejemplo, si el origen es Kevin Bacon,

Escribe actor destino: Jane Fonda

Jane Fonda tiene un número de Bacon de 2

```
Kevin Bacon  
  was in  
Where the Truth Lies Harmon (2005)  
  with  
Deborah Grover  
  was in  
Agnes of God (1985)  
  with  
Jane Fonda
```

Escribe actor destino:

La segunda función recibe como argumento el nombre de un actor (el origen) y muestra para cada número de Bacon cuántos actores hay en la base de datos con ese número, como se muestra a continuación:

Origen: Kevin Bacon

Número	# de actores
0	1
1	1324
2	71566
3	41279
4	1247
5	53
6	0
7	0
8	0
9	0
>9	0