



UNIVERSIDAD DE ALMERÍA

Estructuras de datos y Algoritmo I



Práctica de clase:04

Curso: 2º Grado en Ingeniería Informática

Grupo docente: B1 Grupo: GTA3

Nombre y Apellidos:

Jefferson Max Tomalá Villarreal

PRÁCTICA 4

Aplicaciones de grafos en Java. Red de carreteras

- 1) Además, se deberá desarrollar una memoria que consistirá en un archivo PDF con nombre **practica04**. Para la resolución de esta práctica se ha propuesto utilizar la estructura `TreeMap<K,V>` como estructura de datos para representar un grafo (en este caso un grafo no dirigido y valorado). ¿Sería muy complicado realizar la misma implementación con `HashMap<K,V>`? ¿Qué habría que hacer? ¿Qué diferencias fundamentales existen entre las estructuras `TreeMap<K,V>` y `HashMap<K,V>`?

¿Qué diferencias fundamentales existen entre las estructuras `TreeMap<K,V>` y `HashMap<K,V>`?

El nombre HASH, hace referencia a una técnica de organización de archivos llamada hashing o "dispersion" en el cual se almacenan registro en una dirección del archivo que es generada por una función que se aplica sobre la llave del registro.

El HashMap posee un espacio de memoria, y cuando se guarda un objeto allí, se determina su dirección aplicándole una función a la llave que le indicamos. Por ejemplo:

```
HashMap contenedor = new HashMap();  
Object objeto = new Object();  
String llave = new String("identificador de objeto");  
contendor.add(llave,objeto);
```

Por otro lado, TreeMap implementa un árbol Rojo-Negro cuyas claves están ordenadas.

¿Sería muy complicado realizar la misma implementación con `HashMap<K,V>`? ¿Qué habría que hacer?

No sería complicado utilizar debido a que la funcionalidad de HashMap es igual que TreeMap en la que existe una clave única con un valor asociado o estructura asociada por lo que el sistema a utilizar es el mismo. Para realizar el cambio habría que cambiar el atributo.

adjacencyMap : `TreeMap<Vertex, TreeMap<Vertex, Double>>`

por el siguiente:

adjacencyHash : `HashMap<Vertex, HashMap<Vertex, Double>>`