

Taller 4

Algoritmos de Búsqueda

Árboles rojo-negros / Tablas de dispersión

En procesos de análisis de datos comúnmente se requiere consolidar datos provenientes de múltiples orígenes y realizar distintos tipos de búsquedas y operaciones sobre estos datos.

Como un ejemplo ilustrativo se considerarán los datos históricos de valor de distintas criptomonedas. La siguiente página ofrece acceso a estos datos en forma gratuita:

<https://www.cryptodatadownload.com/data/>

Para descargar los datos ir a la sección “exchanges” y seleccionar uno o varios exchanges sobre los que se podrán descargar datos correspondientes a “pares”. Por ejemplo el par Bitcoin/Dólar se representa BTC/USD y se denota por el símbolo “BTCUSD”.

Para efectos de la realización del ejercicio, descargar al menos los datos correspondientes de tres pares, pero el ejercicio debe ser genérico de forma que le permita al usuario cargar la información de cualquier conjunto de pares.

Ejercicios a desarrollar

1. Implementar un ADT PairData que represente un registro de la información de un par para una fecha determinada. El constructor debe permitir crear una instancia a partir de una línea (separada por comas) del archivo y proveer métodos accesorios. (El dato es inmutable por lo que no se necesitan métodos mutadores).

La clase principal del programa se llamará Taller4 y dentro de esta clase se implementarán las siguientes funcionalidades:

2. Se desea tener la información de los distintos pares en una tabla de símbolos indexada por el nombre del símbolo, de manera que se puede recuperar eficientemente la información correspondiente a un símbolo. Adicionalmente, implementar un método para cargar los datos correspondientes a un par y agregarlos a esta estructura:

```
void cargarDatos(String nombreArchivo)
```

Los datos correspondientes a un par, deben a su vez indexarse por fecha para permitir consultar eficientemente datos en un rango de fechas.

3. Implementar un método para realizar un comparativo del valor por fecha de dos símbolos que se indiquen. Se debe imprimir para fecha tres columnas: La fecha, el valor del primer par y el valor del segundo par, para el rango de fechas especificado.

```
void comparativo(String simbolo0, String simbolo1, Date fecha0, Date fecha1)
```

(Si los símbolos indicados no se encuentran en la tabla, mostrar un mensaje de error amigable)

4. El [coeficiente de correlación de Pearson](#) es un indicador estadístico de la dependencia lineal entre dos variables. Dados dos símbolos y un rango de fechas interesa calcular (eficientemente) la correlación de Pearson entre dos pares:

```
double correlation(String simbolo0, String simbolo1, Date fecha0, Date fecha1)
```

(Si los símbolos indicados no se encuentran en la tabla, mostrar un mensaje de error amigable)

5. Determinar el orden del tiempo necesario para calcular la correlación de Pearson usando el método del punto 4. Contabilizar el tiempo de acceso a las estructuras y el número total de multiplicaciones.

6. El método main debe ofrecer al usuario la posibilidad de indicar el nombre de los archivos a cargar y de ejecutar los procesos de los puntos 3 y 4.

Entregables

Entregar un comprimido con las fuentes del ejercicio. Se puede trabajar en equipos de máximo dos personas. Nombrar el archivo en función de los integrantes, e.g. Apellido1Nombre1-Apellido2Nombre2-Taller4.zip. Incluir también el análisis solicitado en el punto 5.