Tarea 2: *Diff* para archivos

Profesores: Nelson Baloian
Patricio Poblete
Auxiliares: Manuel Cáceres
Sebastián Ferrada
Sergio Peñafiel

Fecha de entrega: 15 de abril, 23:59 hrs

1. Introducción

La distancia de *Levensthein* es una métrica para *strings* que indica la similitud que existe entre dos cadenas de texto. Informalmente, se puede decir que la distancia de Levensthein mide la mínima cantidad de ediciones puntuales que hay que realizar sobre un *string* para convertirlo en el otro. Las operaciones permitidas son: inserción, borrado o sustitución de un caracter. Por lo tanto, una baja distancia de Levensthein indica que los *strings* son muy parecidos y una alta distancia indicará que son muy distintos. Esta distancia también es conocida como distancia de edición entre dos *strings*

Matemáticamente, la distancia de Levensthein entre dos *strings* a y b de largos n y m respectivamente, se denota como $\text{lev}_{a,b}(n,m)$ y se calcula recursivamente de la siguiente manera:

$$\text{lev}_{a,b}(i,j) = \begin{cases} \text{máx}(i,j) & \text{si } \text{mín}(i,j) = 0 \\ \text{mín} \begin{cases} \text{lev}_{a,b}(i-1,j)+1 \\ \text{lev}_{a,b}(i,j-1)+1 \\ \text{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{si no} \end{cases}$$

En donde $1_{(a_i \neq b_j)}$ es la función indicatriz, que toma el valor 1 si $a_i \neq b_j$ y 0 si no, y $\text{lev}_{a,b}(i,j)$ corresponde a la distancia entre los primeros i caracteres de a y los primeros j de b. Notar que el primer elemento en la función mínimo corresponde al caso en que debiera realizarse un borrado de a a b; el segundo, cuando hay una inserción de a a b y el tercero corresponde a si hay que hacer una sustitución o no, dependiendo si los caracteres son el mismo o no.

Por ejemplo, la distancia entre `mojados` y `moteado` es 3:

1. mojados → mojado_ (eliminar **s**)
2. mojado → mo**t**ado (sustituir **j** por **t**)
3. mo**t**ado → mo**te**ado (insertar **e**)

2. Implementación

Lo que se requiere que usted realice en esta tarea es un similar al comando `diff` de Unix que muestra las lineas diferentes entre dos archivos. Para esto, usted debe crear su propia implementación de la función de distancia de Leversthein, pero modificada para archivos. Es decir, usted debe

encontrar cuáles líneas de un archivo deben borrarse, insertarse o sustituirse para ser idéntico a otro.

Como podrá imaginar, una solución que utilice fuerza bruta es bastante lenta y, por lo tanto, no es aceptable. Entonces, se requiere que usted implemente una versión que utilice un enfoque de programación dinámica para resolver este problema.

Cuadro 1: Archivos de Ejemplo

Pollo Perro	Pato Perro Conejo
A.txt	B.txt

Si consideramos los archivos de la Figura 1, notaremos que la distancia de Leversthein entre A.txt y B.txt es 2, ya que hay que cambiar *Pollo* por *Pato* y agregar la línea *Conejo*. En el Listado 1 se puede encontrar un ejemplo de uso del programa y el `output` esperado.

Listing 1: Output esperado

```
>java Diff A.txt B.txt
>1. Pollo       -> Pato
 3.             -> Conejo
 Distance = 2
```

En definitiva, usted debe entregar un archivo `Diff.java` cuyo método `main` reciba dos nombres de archivos como argumentos (Ojo, no leídos desde el teclado) y calcule las ediciones que hay que realizar para que el primero sea idéntico al segundo y muestre la distancia de Levensthein correspondiente.

3. Condiciones de Entrega

- La tarea debe programarse en Java.
- Es obligatorio la entrega de un informe en formato pdf junto con su tarea (Ver siguiente sección).
- Esta tarea es de carácter individual, cualquier caso de copia se evaluará con la nota mínima.
- No olvide subir a U-cursos todos los archivos necesarios para que su tarea funcione correctamente.
- Debe subir los archivos de código fuente (*.java). Los archivos compilados (*.class) no serán evaluados.
- Cualquier duda respecto a la tarea puede ser consultada usando el foro del curso.

- **NO** se aceptarán atrasos.

4. Informe

El informe debe describir el trabajo realizado, la solución implementada, los resultados obtenidos y las conclusiones o interpretaciones de estos. Principalmente debe ser breve, describiendo cada uno de los puntos que a continuación se indican:

- **Portada:** Indicando número de la tarea, fecha, autor, email, código del curso, etc.
- **Introducción:** Descripción breve del problema y su solución.
- **Análisis del Problema:** Exponga en detalle el problema, los supuestos que pretende ocupar, casos de borde y brevemente la metodología usada para resolverlo.
- **Solución del Problema:** Indique claramente los pasos que siguió para llegar a la solución del problema. Muestre mediante figuras y ejemplos qué es lo que realiza su código. Evite copiar todo el código fuente en el informe, sin embargo, puede mostrar las partes relevantes de éste.
- **Resultados y Conclusiones:** Muestre ejemplos de entradas y salidas de su programa, incluyendo capturas de pantalla de estos. Analice la complejidad computacional de su solución. ¿Es mejor que fuerza bruta? ¿Es óptima? Discuta estos resultados y elabore conclusiones sobre su trabajo.