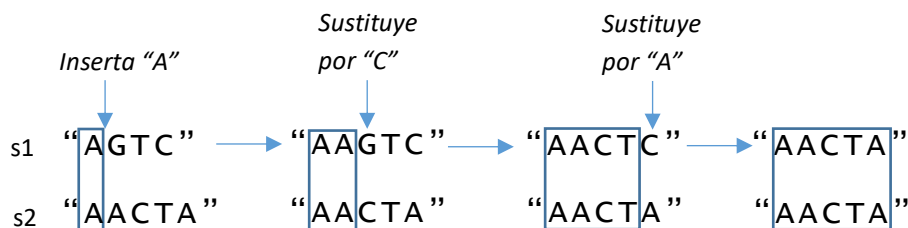


Práctica 2. Módulo Backtracking.

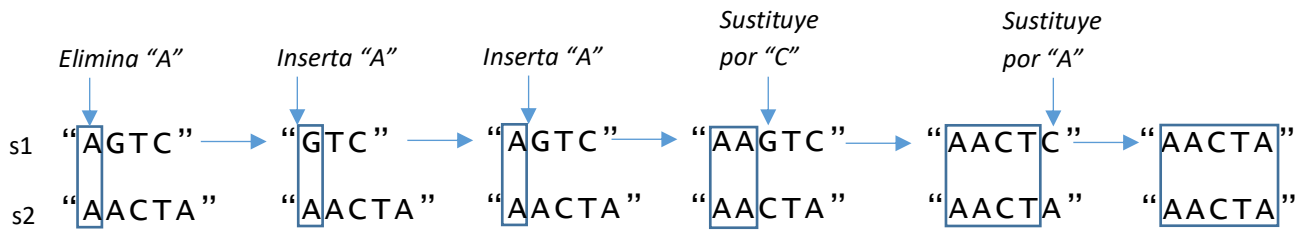
La empresa bioinformática “Virus Fulminator S.A.” está especializada en el estudio genético de los virus buscando como objetivo su erradicación completa de la faz de la tierra. Para llevar a cabo este trabajo, disponen de todo el material necesario para realizar la secuenciación de genomas. Una secuencia de ADN es una cadena ordenada de bases nitrogenadas o nucleótidos (Adenina -A-, Guanina -G-, Citosina -C- y Timina -T-) que codifican la información genética del virus objeto de estudio. Un ejemplo de secuencia de ADN podría ser: AATTTGGCTTTGGAC.

En general, los virus son organismos que se caracterizan por mutar rápidamente. Una mutación produce un cambio permanente en el ADN del organismo, con el objetivo de desarrollar resistencia a un medicamento, o aumentar la tasa de infectividad, entre otros. Normalmente las mutaciones de los virus suelen producirse en zonas de su ADN que no ponen en peligro su existencia. Una de las estrategias seguidas por la empresa para eliminar virus se basa en comparar las secuencias de ADN de muestras de distintas poblaciones del mismo virus, con el fin de identificar las zonas estables del genoma y atacarlas con nuevas vacunas.

Para realizar esta labor, en la empresa “Virus Fulminator S.A.” precisan de un algoritmo que permita comparar dos secuencias de ADN, s_1 y s_2 y determine el número mínimo de mutaciones que deben producirse en la secuencia s_1 para que se transforme en la secuencia s_2 , teniendo en cuenta que s_1 y s_2 pueden tener distinta longitud. Las mutaciones que interesa considerar son aquellas producidas por sustitución (un nucleótido cambia por otro nucleótido y la cadena resultante sigue teniendo la misma longitud), eliminación (se elimina un nucleótido y la cadena resultante se acorta en una unidad) e inserción (se inserta un nuevo nucleótido y la cadena resultante se incrementa en una unidad). Por ejemplo, dadas las secuencias $s_1 = \text{“AGTC”}$ y $s_2 = \text{“AACTA”}$, el número mínimo de mutaciones necesarias para transformar s_1 en s_2 es 3:



El número de combinaciones de mutaciones para transformar s_1 en s_2 es muy grande, pero a la empresa sólo le interesa obtener la mínima. Por ejemplo, la siguiente secuencia de mutaciones:



también transforman la secuencia s1 en s2, pero en 5 mutaciones.

SE PIDE:

- Implementar un algoritmo en Java, basado en **Selección Óptima**, que resuelva el problema planteado.

Nota: Es requisito para la corrección de la práctica que el programa entregado compile sin errores y funcione correctamente. De igual manera, es imprescindible que el esquema implementado sea el de Selección Óptima y se incorpore el concepto de poda (en el código desarrollado debe quedar claro: los movimientos o candidatos que se consideran en cada nodo del árbol de estados, la condición de aceptable de un movimiento, la condición de solución, las operaciones de anotar y desanotar y la condición de poda).

DESARROLLO:

Se proporcionan dos clases: *Pruebas* (clase lanzadera para probar el algoritmo) y *Principal* (clase donde hay que implementar el algoritmo pedido).

Pruebas.Java

Esta clase proporciona una lanzadera para realizar las pruebas con las que poder verificar que la codificación realizada por el alumno cumple con los requisitos.

Principal.Java

Esta clase contiene el método que se tiene que implementar.

```
public static int numMinMutaciones(String cadena1, String cadena2)
```

donde:

- El parámetro String **cadena1**, especifica la secuencia de ADN a transformar.
- El parámetro String **cadena2**, especifica la secuencia de ADN objetivo.

El algoritmo debe devolver el número mínimo de mutaciones que deben realizarse en cadena1 para transformarla en cadena2.

Nota: Se recomienda consultar la documentación de la clase String del API de Java (paquete java.lang): <https://docs.oracle.com/javase/8/docs/api/>

NORMAS DE ENTREGA:

La entrega consistirá en subir a la plataforma Moodle un archivo en formato comprimido (ZIP o RAR) que incluya el fichero Principal.java donde se encuentre implementado el algoritmo.

Si la práctica se ha desarrollado individual, el nombre del fichero comprimido deberá ser el número de matrícula del alumno (ejemplo: ai0260.zip). Si la práctica se ha desarrollado en pareja con otro compañero, el nombre del fichero comprimido deberá ser el número de matrícula de ambos (ejemplo: ai0340-bc0433.zip). En este último caso, **AMBOS** alumnos deberán subir la práctica al Moodle.

El archivo comprimido se deberá subir a la plataforma Moodle **antes de las 23:55 horas del 27 de octubre de 2019**. Aquellos alumnos que hayan subido la práctica deberán realizar un examen escrito de la misma el viernes 8 de noviembre de 2019. Para poder aprobar esta práctica será indispensable haberla subido a la plataforma, haber superado el examen de la misma y haberla desarrollado correctamente.