

Primer Examen de Programación  
Curso 2018-2019

## Estabilizar Genes

**NOTA:** Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

En el Centro de Ingeniería Genética y Biotecnología (CIGB) se llevan a cabo numerosas investigaciones vinculadas al código genético de seres vivos. Un gen se define por la secuencia de bases nitrogenadas que lo componen, las cuales pueden ser de cuatro tipos: adenina(A), timina(T), citosina(C) y guanina(G).

En investigaciones científicas realizadas se han detectado secuencias inestables de ADN en algunas especies. Dichas secuencias implican cambios significativos en el desarrollo de los organismos que las presentan y resulta de interés para el centro la estabilización de esas secuencias para preservar el equilibrio biológico.

Un gen es estable si está compuesto por la misma cantidad de bases nitrogenadas y todas están presentes. Esto es equivalente a que cada base se repita exactamente  $n/4$  veces en la cadena. Por ejemplo, los genes siguientes son estables:

Ejemplo 1

G A C T

Ejemplo 2

A A G T G C C T

El proceso de modificar un gen es extremadamente costoso por lo que es necesario realizar la menor cantidad de cambios posibles. Adicionalmente, solo se puede modificar una subcadena en cada gen. En los siguientes ejemplos se muestran secuencias inestables de ADN y sus soluciones óptimas.

Ejemplo 1

Gen original

T T C G

Gen modificado

A T C G

Ejemplo 2

G A A A T A A A

G A G C C T T A

Note que el gen estable óptimo puede no ser único. En el caso de la cadena "TTCC" son posibles genes estables las cadenas "ATCG" y "TACG" y la subcadena óptima a reemplazar es "T". Mientras que para la cadena "GAAATAAA" cadenas óptimas son "GAGCCTTA", "GGCCTTAA", "GACCTTGA", entre otras. Por otro lado, siempre es posible obtener una cadena estable.

Teniendo en cuenta lo anteriormente planteado, su tarea consiste en implementar un algoritmo que reciba un gen representado como una cadena de caracteres y devuelva la menor subcadena que debe ser reemplazada para hacerlo estable. En el ejemplo 1 la salida es "T" y en el ejemplo 2 una posible salida podría ser "AATAA".

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe implementar el método `EstabilizaGen` que se encuentra en la clase `EstabilizarGenes` en el *namespace* `Weboo`. Examen. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class EstabilizarGenes
    {
        public static string EstabilizaGen(string gen)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

Este método tiene como primer parámetro un `string` que representa la cadena de ADN que se desea hacer estable. Usted debe implementar este método de forma tal que se devuelva finalmente un `string` con la menor subcadena que, al ser **reemplazada**, convierte a la cadena original en estable.

Puede asumir que:

- El tamaño `n` de la cadena de bases nitrogenadas que representa al gen siempre es múltiplo de 4 y mayor que 0, es decir, `n > 0` y `n % 4 == 0`.

**NOTA:** Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará. En particular, es imprescindible que usted no cambie el parámetro del método `EstabilizaGen`. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

## Ejemplos

```
//EJEMPLO 1
string gen1 = "TTCG";
string subString1 = EstabilizarGenes.EstabilizarGen(gen1);
//Respuesta correcta "T"

//EJEMPLO 2
string gen2 = "GAAA";
string subString2 = EstabilizarGenes.EstabilizarGen(gen2);
//Respuesta correcta "AA"

//EJEMPLO 3
string gen3 = "GAAATAAA";
string subString3 = EstabilizarGenes.EstabilizarGen(gen3);
//Respuesta correcta "AATAA"

//EJEMPLO 4
string gen4 = "ACTGCTAG";
string subString4 = EstabilizarGenes.EstabilizarGen(gen4);
//Respuesta correcta ""
```

```
//EJEMPLO 5
string gen5 = "ACTGAAAG";
string subString5 = EstabilizaGenes.EstabilizarGen(gen5);
//Respuesta correcta "AA"
```

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.