INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

Fundamentos Matemáticos de la Computación

Proyecto Final

Complejidad de Kolmogorov

“Manual técnico”

Equipo 5

Integrantes

Susana Muñoz Acosta 165889

Manuel Fernandez Verda 166496

Sebastián Valderrábano Cabrera 163791

Rubén Romero Ortega 174178

13 de mayo de 2020

Introducción

Para la creación de nuestro proyecto, el equipo se basó en un set de códigos elaborados por el profesor Kuri. En específico, se modificó el código RMF, el cual contiene 33 diferentes problemas de optimización que se resuelven utilizando algoritmos evolutivos. Los primeros 32 problemas fueron planteados por Kuri y el problema 33 fue desarrollado para calcular la complejidad de Kolmogorov.

También se modificó la clase RMH20, que contiene las funciones necesarias para ejecutar las funciones RMF e imprimir los resultados. Se implementaron y modificaron funciones para que el display de los resultados en el caso de que se eligiera resolver la función Kolmogorov se muestre cómo se veía la máquina de Turing elegida.

Código

En esta sección se muestra el código que desarrollamos para la resolución del problema y el cálculo de la complejidad de Kolmogorov.

Class RMF

Para ello se agregó a la clase RMF la función ”Kolmogorov” que recibe el genoma generado en la clase RMH20a y la cinta meta que es decodificado del archivo elegido. Además de esto se agregó al switch de la función “Evalúa” en RMH20a el caso 33 donde se mandaba a llamar esta función.

Kolmogorov inicializa los valores necesarios para llamar el análisis completo de Kolmogorov en la clase UTM. Para resolver la Máquina de Turing se utilizó un número máximo de iteraciones de 1,000,000, una cinta inicial de 10,000 ceros y la posición inicial a la mitad de la cinta, es decir en la posición 5,000.

Cuando se termina de calcular el resultado de la máquina de Turing, se calcula el error entre el resultado y la cinta meta. El resultado de esta función regresa un arreglo de 4 números flotantes: el número máximo de coincidencias, el error porcentual con respecto a la cinta meta, la posición inicial donde se encontró el mayor número de coincidencias y la longitud de la meta.

```
public static double[] Kolmogorov (String genoma, String goal) throws Exception {

    //Se inicia con una cinta de 10,000 ceros

    //Número máximo de iteraciones son 1,000,000
```

```
    //Posición inicial de la cinta es 5000
      String Cinta="";
      int N=1000000,
      posInit=5000;
      for(int i=0;i<10000;i++){
         Cinta=Cinta+"0";
      }
      //Se crea la MT y se guarda su resultado
      String[] out=UTM.NewTape(genoma, Cinta, N, posInit);
          //Se ejecuta la función de error de Kolmogorov con los parámetros: goal,Resultado TM, PosInicialTM, PosFinalTM
      return UTM.Kolmogorov(goal,out[0],out[3],out[4]);
  }
```

Class UTM

De igual manera se modificó la clase UTM, se agregó la función “Kolmogorov” y se modificó la función “New Tape”.

Originalmente, la cinta New Tape regresaba la cinta resultante de la Máquina de Turing, no obstante, para facilitar el procedimiento de la complejidad de Kolmogorov, se decidió implementar cambios.

La función NewTape cambia la cinta y la Máquina de Turing binaria ingresada a arreglos para facilitar su manejo. Con los arreglos definidos, recorre la máquina de Turing binaria y escribe en el arreglo de la cinta conforme avanzan las iteraciones, también se guarda en un arreglo de 64 posiciones los estados visitados. Al terminar el número máximo de iteraciones o pasar el tamaño de la cinta inicial, sale del ciclo y convierte los resultados en cadenas de caracteres para regresarlos. La función NewTape regresa como resultado un arreglo de Strings que contiene: la cinta modificada, el número de estados utilizados, una cadena binaria de 64 posiciones que

determina qué estados se ocuparon, la posición mínima y la posición máxima en la que se accedió para escribir en la cinta.

NEW TAPE

```
public static String[] NewTape(String TT, String Cinta, int N, int posInit)
  {
    int currentState=0; //Estado actual
    int iteraciones=0; //Contador para número de iteraciones
    int[] base=new int[Cinta.length()]; //Cinta
    //Definimos la posición máxima y mínima de escritura
    int posMin=posInit;
    int posMax=posInit;
    //Arreglo de Strings para regresar la información
    String[] resultado=new String[5];

    //Se llena el arreglo base con la cadena Cinta.
    //En el arreglo es más sencillo manejar posiciones y cambiar valores.
    for (int c=0;c<Cinta.length();c++){
      base[c]=Integer.parseInt(Cinta.charAt(c)+"");
    }

    String[] states=new String[TT.length()/8]; //TT

    //Arreglo que marcará los estados utilizados durante el procedimiento
```

```
int[] nstates=new int[64];
for (int c=0; c<64;c++){
    nstates[c]=0;
}
```

//Se lleva el arreglo states con las definiciones de estados, estas definiciones tienen 8 bits

```
for (int c=0;c<TT.length()/8;c++){
```

states[c]=TT.substring(c*8, c*8+8); **//Se toma el substring de 8 bits, los parámetros de substring son de tipo [a,b)**

```
}
String aux,res; //variables auxiliares
```

while (currentState!=63 && iteraciones<=N){ **//while el estado final no sea HALT y el número de iteraciones no se pase del límite**

```
    iteraciones++; //aumentamos iteraciones
```

if(posInit<0 || posInit>=base.length){ **//Si nos pasamos del tamaño de la cinta, nos salimos**

```
        break;
    }
    //Anotar que ya pasamos por este estado
    nstates[currentState]=1;//////////////////////////////////////////////////////////////////////////////
```

//Estado se encontrara en states[currentState*2] y states[currentState*2+1]

```
    if(base[posInit]==0){
        aux=states[currentState*2];
```

```
        }
        else{
            aux=states[currentState*2+1];
        }
        //Escribimos el valor en la posInit
        base[posInit]=Integer.parseInt(aux.charAt(0)+"");
        //Si cambiamos de valor 0 a 1 o 1 a 0 la productividad se ve afectada
        //if(flag && aux.charAt(0)=='1'){productividad++;}
        //if(!flag && aux.charAt(0)=='0'){productividad--;}

          //Cambiamos la posicion del arreglo dependiendo del segundo caracter. 0
D, 1 I
        if(aux.charAt(1)=='0'){
            posInit++;
            if(posInit>posMax && posInit < base.length)
                posMax=posInit;
        }
        else{
            posInit--;
            if(posInit<posMin && posInit >= 0)
                posMin=posInit;
        }
```

```
            //Calculamos el siguiente estado convirtiendo el número de base 2 a base
10
            currentState=Integer.parseInt(aux.substring(2,8),2);
        }
        //Guardamos una cadena de caracteres con los resultados
        res="";
        for (int i=0;i<base.length;i++ )
            res+=base[i];
        int numEstadosUsados=0;
        //Condiciones finales: Se alcanza HALT
        if(iteraciones<=N && posInit<base.length){
            //System.out.println("HALT state was reached");
            //System.out.println("Total of "+iteraciones+" transitions");
            resultado[2]="1";
        }
        //Condiciones finales: No se alcanza HALT
        else{
            //System.out.println("HALT state was not reached");
            resultado[2]="0";
        }

        //Se guarda una cadena binaria para marcar los estados utilizados
        String turingMachineAcotada="";
```

```
        for (int k=0;k<64;k++){
            if(nstates[k]==1){
                    numEstadosUsados++; //Si se utilizó, aumentamos el num de estados usados
                turingMachineAcotada+="1";
            }
            else{
                turingMachineAcotada += "0";
            }
        }
        resultado[0]=res; //Cadena resultante
        resultado[1]=numEstadosUsados+""; //NumEstadosUsados
        resultado[2]=turingMachineAcotada; //Cadena binaria con edos usados
        resultado[3]=posMin+""; //Posicion minima alcanzada
        resultado[4]=posMax+""; //Posicion máxima alcanzada
        return resultado;
    }
```

Por su parte, la función Kolmogorov recibe la cinta meta, así como las posiciones inicial y final en la que la Máquina de Turing escribió y la cadena resultante. Para realizar la comparación, fue necesario limitar la cinta únicamente a lo que realmente cambió la Máquina de Turing con respecto a la cinta inicial de ceros, pues es la mejor forma de buscar qué Máquina de Turing replica la cinta meta.

La función Kolmogorov primeramente recorta la cinta resultado de la Máquina de Turing, después comienza una comparación posicional entre la cinta recortada y la cinta meta. Esta comparación busca el mayor número de coincidencias, por lo que se

va recorriendo la posición inicial de análisis de la cinta recortada buscando a partir de qué posición se obtiene un resultado mejor.

Cada vez que se registra un mejor número de coincidencias, se guarda la posición inicial a partir de cuál se realiza la comparación. La función Kolmogorov regresa como resultado un arreglo de números flotantes que incluye: el número máximo de coincidencias encontrado, el error porcentual de diferencia, la posición mínima y la longitud de la meta para comparar los datos en el análisis del genoma.

KOLMOGOROV

```
public static double[] Kolmogorov(String goal, String tM, String posInit, String posFinal){
    int matches; //Número de valores encontrados en la cinta resultado de la MT
    int posMin=0;  //Posición Mínima para el análisis
    //Aquí guarda la cadena que realmente escribió la TM
    String tMres=tM.substring(Integer.parseInt(posInit), Integer.parseInt(posFinal)+1);
    //Inicialización de variables de posición
    int posInicialTM=0;
    int posInicialRes=0;
    int bestMatch=0;
    //Si la cinta de TM es mayor que la de la meta, se recorrerá una por una hasta encontrar el mayor número de matches
    if(tMres.length()-goal.length()>=0){
        for(int i=0;i<tMres.length()-goal.length();i++){
            posInicialTM=i;
            matches=0;
            posInicialRes=0;
            while(posInicialRes<goal.length()){
```

```
            if(tMres.charAt(posInicialTM)==goal.charAt(posInicialRes))
                matches++;
            posInicialTM++;
            posInicialRes++;
        }
        //Actualizamos información si se logró un mejor resultado
        if( matches>bestMatch){
            posMin = posInicialTM-goal.length(); //Guardamos donde empieza el valor
            bestMatch=matches;
        }
        if(bestMatch == goal.length()){
            break;
        }
    }
}
//Si la cinta de TM es menor, solo checamos una vez
else{
    while(posInicialTM<tMres.length()){
        if(tMres.charAt(posInicialTM)==goal.charAt(posInicialRes))
            bestMatch++;
        posInicialTM++;
        posInicialRes++;
```

```
        }
    }
    double[] res=new double[4];
    res[0]= bestMatch; //Número maximo de matches
    res[1]=1-(double)bestMatch/goal.length(); //error
    res[2] = posMin; //Posición donde se inicia
    res[3] = goal.length(); //Longitud del goal


    return res;
}
```

La función Imprime recibe la Máquina de Turing completa, la cadena binaria que denota los estados utilizados y el número de estados que se utilizaron. Esta función imprime en formato de tabla legible la Máquina de Turing sólo con sus estados que realmente se utilizaron.

IMPRIME

```
public static void imprime(String TT, String edos, int numEdos){
    int NumStates = TT.length()/ 16;
    int ix16,
    x0_I,
    x1_I,
    Estado;
    String x0_M,
    x1_M;
    //Se imprimen solo los estados utilizados
```

```
System.out.println("Hay " + numEdos + " estados en la Maquina de Turing");
System.out.println("****************************************");
System.out.println("Complejidad de Kolmogorov: "+ numEdos*16);
System.out.println("****************************************");
System.out.println("");
 System.out.println(" EA | O | M | SE || O | M | SE |");//Estado Actual | Escribe | Se mueve |Siguiente Estado
System.out.println(" ------------------------------");
for (int i = 0; i < NumStates; i++) {
   if(edos.charAt(i)=='1'){ //Si el estado no fue utilizado, se salta esta sección
      System.out.printf("%4.0f|", (float) i);
      ix16 = i * 16;
      x0_I = Integer.parseInt(TT.substring(ix16, ix16 + 1));
      x0_M = TT.substring(ix16 + 1, ix16 + 2);
      if (x0_M.equals("0")) x0_M = " R |";
      else x0_M = " L |";
      System.out.printf("%3.0f|" + x0_M, (float) x0_I);
      Estado = 0;
      for (int j = ix16 + 2; j < ix16 + 8; j++) {
            Estado = Estado * 2;
            if (TT.substring(j, j + 1).equals("1")) Estado++;
            //endif
      } //endFor
```

```
        if (Estado == 63) System.out.print("   H||");
        else System.out.printf("%4.0f||", (float) Estado);
        //endif
        x1_I = Integer.parseInt(TT.substring(ix16 + 8, ix16 + 9));
        x1_M = TT.substring(ix16 + 9, ix16 + 10);
        if (x1_M.equals("0")) x1_M = " R |";
        else x1_M = " L |";
        System.out.printf("%3.0f|" + x1_M, (float) x1_I);
        Estado = 0;
        for (int j = ix16 + 10; j < ix16 + 16; j++) {
            Estado = Estado * 2;
            if (TT.substring(j, j + 1).equals("1")) Estado++;
            //endif
        } //endFor
        //en el estado Halt se escribe “H”
        if (Estado == 63) {
            System.out.print("   H|\n");
        } else {
            System.out.printf("%4.0f|\n", (float) Estado);
        } //endif
    }
} //endFor
```

```
}
```

Class RMH20a

Finalmente, se implementó una función de lectura en el main para registrar la cinta meta desde un documento de texto.

LECTURA

```
public static void lectura() throws IOException {
    //Lee el archivo para el cálculo de Kolmogorov
    BufferedReader Kbr;
    Kbr = new BufferedReader(new InputStreamReader(System.in));
    String sResp;
    RandomAccessFile Datos = null;
    String aux = "";
    String aux2 = "";
    String aux3 = "";
    Boolean flag = false;

    int i;
    boolean ask = false;
    while (true) {
       if (ask) { // No preguntes la primera vez
          System.out.println("\nDesea leer otro archivo?");
          System.out.println("\"S\" para continuar; otro para terminar...)");
```

```
            sResp = Kbr.readLine().toUpperCase();
            if (!sResp.equals("S")) {
                break;
            }
            //endIf
        } else {
            ask = true; // Pregunta de la segunda en adelante
        } //endif
        System.out.println("Deme el nombre del archivo de datos que quiere leer:");
        String FName = Kbr.readLine().toUpperCase();
        try {
            //Abre el documento y lo convierte de ASCII a binario si encuentra un caracter distinto de 0 ó 1
            File file = new File(FName);
            Scanner sc = new Scanner(file);
            while (sc.hasNextLine()) {
                aux += sc.nextLine();
            }
            for (int j = 0; j < aux.length(); j++) {
                aux2 += aux.charAt(j);
                aux3 += Integer.toBinaryString(Character.getNumericValue(aux.charAt(j)));
                if (aux.charAt(j) != '0' || aux.charAt(j) != '1') {
                    flag = true;
```

```
                }
            }
            //Actualiza el valor del flag
            if (flag) {
                goal = aux3;
            } else {
                goal = aux2;
            }
            //System.out.println(goal);
            break;
        } //endTry
        catch (Exception e1) {
            System.out.println("No se encontro \"" + FName + "\"");
        } //endCatch
    }

}
```

Pruebas

En esta sección se incluyen 5 pruebas que se realizaron bajo las condiciones iniciales de la Máquina de Turing. Los resultados se muestran exactamente como los imprime la terminal de Java. Para los 5 casos se utilizó un total de 3,000 iteraciones máximas, y una longitud de 1023 bits en el genoma.

Prueba 1:

Cadena meta:
1101110111011101110111011101110111000001110111011101110111011101110111011101110111011
1011101110111011101110111011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101110111011101110111011101
1101110111011101110111011101110111011101110111011101110111011101110111011101110111011
1011101110111011101110111011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101110111011101110111011101
1101110111011101110111011101110111011101110111011101110111011101110111011101110111011
101110111011101110111011101110111011101110111011101110111011101110111011101110111

Ejecución:

1) Funcion a optimizar: 33

2) Bits para enteros: 1023

3) Bits para decimales: 0

4) Numero de variables: 1

** Long. del genoma: 1024

5) Numero de iteraciones: 3000

6) Minimiza[0]/Maximiza[1]: 0

Modificar (S/N)? N

Deme el nombre del archivo de datos que quiere leer:

RESMT1.txt

BEST new error: 0.7451274362818591 Tamaño: 667.0 Matches: 170.0 Pos Inicial: 0.0

BEST new error: 0.7436281859070465 Tamaño: 667.0 Matches: 171.0 Pos Inicial: 4333.0

BEST new error: 0.7421289355322338 Tamaño: 667.0 Matches: 172.0 Pos Inicial: 0.0

BEST new error: 0.25337331334332835 Tamaño: 667.0 Matches: 498.0 Pos Inicial: 0.0

BEST new error: 0.005997001499250421 Tamaño: 667.0 Matches: 663.0 Pos Inicial: 2.0

BEST new error: 0.004497751124437732 Tamaño: 667.0 Matches: 664.0 Pos Inicial: 4334.0

5.0E-4 MegaIters

0.001 MegaIters

0.0015 MegaIters

0.002 MegaIters

0.0025 MegaIters

Óptimo: .0044977

Hay 15 estados en la Maquina de Turing

****

Complejidad de Kolmogorov: 240

****

EA	O	M	SE		O	M	SE
0	1	R	15		1	R	61
2	0	L	47		0	L	29
3	1	L	2		1	R	37

15| 0| R | 34|| 1| R | 34|

21| 0| L | 40|| 0| L | 26|

24| 1| L | 3|| 0| R | 19|

26| 0| L | 37|| 0| R | 45|

27| 1| L | 21|| 0| L | 11|

29| 1| R | 50|| 1| R | 26|

34| 1| R | 29|| 1| L | 6|

37| 0| L | 58|| 1| L | 37|

40| 0| R | 24|| 1| R | 26|

47| 1| L | 24|| 0| R | 20|

50| 0| L | 29|| 0| L | 8|

58| 0| R | 44|| 1| L | 27|

Cadena esperada:
1101110111011101110111011101110111000001110111011101110111011
1011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101
1101110111011101110111011101110111011101110111011101110111011
1011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101
1101110111011101111

Cadena encontrada:
1101110111011101110111011101110111011101110111011101110111011
1011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101
1101110111011101110111011101110111011101110111011101110111011
1011101110111011101110111011101110111011101110111011101110111
0111011101110111011101110111011101110111011101110111011101110
1110111011101110111011101110111011101110111011101110111011101
11011101110111011110

Prueba 2:

Cadena meta:

I want you to know one thing. You know how this is:

if I look

at the crystal moon, at the red branch

of the slow autumn at my window,

if I touch

near the fire

Ejecución:

1) Funcion a optimizar: 33

2) Bits para enteros: 1023

3) Bits para decimales: 0

4) Numero de variables: 1

** Long. del genoma: 1024

5) Numero de iteraciones: 3000

6) Minimiza[0]/Maximiza[1]: 0

Modificar (S/N)? N

Deme el nombre del archivo de datos que quiere leer:

RESMT2.txt

BEST new error: 0.8477218225419665 Tamaño: 1668.0 Matches: 254.0 Pos Inicial: 4.0

BEST new error: 0.15107913669064743 Tamaño: 1668.0 Matches: 1416.0 Pos Inicial: 3336.0

5.0E-4 MegaIters

BEST new error: 0.1504796163069544 Tamaño: 1668.0 Matches: 1417.0 Pos Inicial: 0.0

0.001 MegaIters

0.0015 MegaIters

0.002 MegaIters

0.0025 MegaIters

Óptimo: .1504796

Hay 15 estados en la Maquina de Turing

****

Complejidad de Kolmogorov: 240

****

EA	O	M	SE	O	M	SE
0	1	R	60	1	R	43
2	1	R	58	1	R	33
5	0	R	H	0	R	54
10	1	R	53	1	R	34
21	1	L	2	1	R	35
28	0	L	49	1	L	42
33	1	R	44	1	L	28
35	1	R	46	1	R	60
42	1	R	5	0	R	40
46	0	L	21	0	R	29

50| 0| R | 62|| 1| L | 22|

53| 1| R | 35|| 1| L | 35|

54| 1| L | 53|| 0| R | 50|

60| 1| L | 2|| 0| L | 53|

62| 0| L | 10|| 1| L | 19|

Cadena esperada:
10010111111111111111111111111111111000001010101111110111111111111111111111111111111
1111100010110001111011111111111111111111111111111111111101110001111111111111111111111
1111111110100101111000100000111111111111111111111111111111111100010111110111111111
11111111111111111111111111011000110010101111000011111111111111111111111111111111111111
1111111111111111111111111111000101100011110111111111111111111111111111111111101001011
1110001000001111111111111111111111111111111111000111000100000111111111111111111111111
1111111111011000110010111001111111111111111111111111111111100101110011111111111111
111111111111111111100101111111111111111111111111111111111001011111111111111111111111
1111111111010111000110001010010101110111111111111111111111111111111111111011000111
1011111111111111111111111111111111100110111000101110011101101010101111111111111111
111111111111111101101100011000111111111111111111111111111111111111111111111111111111
11111111111111101011101111111111111111111111111111111111110110001110111111111111111
111111111111111110111101111111111111111111111111111111111111101110111010101111001
0001110001111111111111111111111111111111111111101100011101111111111111111111111111
1111111100101011100010000011111111111111111111111111111111101011101110111101011 0
10111111111111111111111111111111111111010111011111111111111111111111111111111111011010
0010111111111111111111111111111111111100000100101011110111000100000111111111111111111
11111111111111111001011111111111111111111111111111111111100101111111111111111111111111
1111111111011100011101100100011011111010101101111111111111111111111111111111111111
1011000111101111111111111111111111111111111111111001011011110

Cadena encontrada:
100111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

Prueba 3:

Cadena meta:

Youth, treasure only gods may keep,

Fleeting from me forever now!

I cannot, when I wish to, weep,

And often cry I know not how…

My heart's celestial histories,

So countless were, could not be told.-

She was a tender child, in this

World of affliction manifold.

She seemed a dawn of pure delight;

She smiled as the flowers after rain;

Her tresses were like to the night

Fashioned of darknesses and pain.

Ejecución:

1) Funcion a optimizar: 33

2) Bits para enteros: 1023

3) Bits para decimales: 0

4) Numero de variables: 1

** Long. del genoma: 1024

5) Numero de iteraciones: 3000

6) Minimiza[0]/Maximiza[1]: 0

Modificar (S/N)? N

Deme el nombre del archivo de datos que quiere leer:

resmt3.txt

BEST new error: 0.9982403217697335 Tamaño: 3978.0 Matches: 7.0 Pos Inicial: 0.0

BEST new error: 0.1483157365510307 Tamaño: 3978.0 Matches: 3388.0 Pos Inicial: 0.0

BEST new error: 0.14781297134238314 Tamaño: 3978.0 Matches: 3390.0 Pos Inicial: 6.0

BEST new error: 0.14731020613373558 Tamaño: 3978.0 Matches: 3392.0 Pos Inicial: 0.0

5.0E-4 MegaIters

0.001 MegaIters

0.0015 MegaIters

0.002 MegaIters

0.0025 MegaIters

Óptimo: .1473102

Hay 28 estados en la Maquina de Turing

****

Complejidad de Kolmogorov: 448

****

EA | O | M | SE || O | M | SE |

0| 1| R | 25|| 1| R | 29|

3| 0| L | 11|| 1| R | 9|

4| 1| L | 55|| 0| R | H|

10| 1| L | 17|| 0| R | 5|

11| 0| R | 21|| 0| R | 45|

12| 0| R | 34|| 0| L | 57|

15| 0| R | 10|| 0| L | 57|

17| 0| L | 61|| 1| L | 11|

20| 0| L | 52|| 1| R | 57|

21| 1| R | 28|| 1| L | 0|

25| 1| L | 58|| 0| L | 31|

27| 1| R | 48|| 1| L | 44|

28| 1| R | 47|| 1| L | 9|

29| 0| L | 2|| 0| R | 44|

33| 1| L | 42|| 1| R | 17|

34| 1| R | 48|| 0| R | 3|

41| 1| R | 41|| 0| L | 20|

42| 0| L | 56|| 0| L | 4|

44| 1| L | 6|| 1| R | 33|

45| 0| L | 4|| 0| R | 41|

47| 1| L | 27|| 1| R | 10|

48| 1| L | 29|| 1| L | 16|

49| 0| R | 51|| 1| R | 12|

52| 1| R | 15|| 1| L | 41|

55| 0| R | 52|| 0| R | 41|

57| 0| L | 41|| 1| L | 9|

58| 1| R | 28|| 0| R | 49|

61| 1| L | 16|| 1| R | 12|

Cadena esperada:

1000101100011110111011000111111111111111111111111111111111111111111111111111111111111111111111
1111111101110111110101011100111101101111101111111111111111111111111111111111111100010111
1010110001011111111111111111111111111111111110000110001101111001111111111111111111111111
1111111111101101010100010111111111111111111111111111111111101001110111011001111111111
1111111111111111111111111101011101110111011001010111100001111111111111111111111111
1111111111111011110001011011111111111111111111111111111111011011101111111111111111
1111111111111111111110001101111101111111101101111111111111111111111111111111101111
1000100000111111111111111111111111111111111100101111111111111111111111111111110010
1010111101111100011101111111111111111111111111111111111111111111111111111111111111
1000001000111101011111111111111111111111111111111111111001011111111111111111111111
1111100000100101110010001111111111111111111111111111111111101110001111111111111111
1111111111111111111111111111111111111111111111111000001110111011001111111111111111
1111111111110101011111011111111111111111111111111111111111000111111101111010111111
1111111111111111111111111111100110111000101111111111111111111111111111111110010111111
1111111111111111111111111010010111110001000001111111111111111111111111111101111
1000111011111111111111111111111111111111111000111000100000111111111111111111111111
1111101101000101111111111111111111111111111111111000111101010110111110111111111111
1111111111111111111100111111111111111111111111111111111001110101011110111001110110
0101010101011111111111111111111111111111111000110010111001110111000110111001011
0111001111111111111111111111111111111111100110001111111111111111111111111111001
1000111101011111101101011110111001110011111111111111111111111111111111000001110110
1111101111111111111111111111111111111111111111111111111111111111111111100110001111010
1011101111111111111111111111111111111101111000111011111111111111111111111111111
1011111011111111111111111111111111111111011100010101110111111111111111111111111111
1111111111111111111111111111111111111111001000111101111111111111111111111111110
0000101011100111111111111111111111111111111101011111111111111111111111111111110
1111010111110111101101111111111111111111111111111111111100100011001010101110111111
1111111111111111111111111111111111111111111111111111111100101011111111111111111111
1111111111111111011000110010111001000001100011011101011101111111111111111111111111
1111111111000111111111111111111111111111111111101011111111010110010110011101100
0110001011111111111111111111111111111111111101101010101111001011111100010101110111
1111111111111111111111111111111001000111101111111111111111111111111111110011101
1101011011101101111111111111111111111111111111101011111111111111111111111111111
1011010100000101111111111111111111111111111111100011111111111111111111111111111
1111111100111110110111110111111111111111111111111111111111101111010101100101000010
0111101111111111111111111111111111111111100100011110111111111111111111111111111111
1100101101001010101111011011111111111111111111111111111111101011100111111111111111

111111111111111111111011000111101111111111111111111111111111111111111010111000100000 1
11011011111001111111111111111111111111111111111101011111101111011011111111111111111111111
11111111111111111011101010010101111111111111111111111111111111111111000111101101111111
111111111111111111111111111111011101111101110011100111011100111111111111111111111111111
111111110000011101101111011111111111111111111111111111111111111111010110010101001110111111
111111111111111111111111111111011100011111111111111111111111111111111111101100011110111
1111111111111111111111111111111110111100101000010001111011111101011100100011001011000
101111110110111111111111111111111111111111111111100011111111111111111111111111111111111
1101101011011101001011111101110011100111011100111111111111111111111111111111111111110101
0111110111111111111111111111111111111111100110101001010111111111111111111111111111111111
111111

Cadena encontrada:
100110010011111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

```
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
111111111111111111111
```

Prueba 4:

Cadena meta:

Para pasar la materia de fundamentos matematicos de la computacion

Realizamos un codigo que aproxima la complejidad de Kolmogorov

Ejecución:

```
1) Funcion a optimizar:          33

2) Bits para enteros:            1023

3) Bits para decimales:          0

4) Numero de variables:          1

** Long. del genoma:             1024

5) Numero de iteraciones:   3000

6) Minimiza[0]/Maximiza[1]: 0


Modificar (S/N)? N

Deme el nombre del archivo de datos que quiere leer:

Resmt4.txt

BEST new error: 0.9971724787935909       Tamaño: 1061.0         Matches: 3.0   Pos Inicial: 0.0
```

BEST new error: 0.7992459943449576 Tamaño: 1061.0 Matches: 213.0 Pos Inicial: 2.0

BEST new error: 0.48539114043355325 Tamaño: 1061.0 Matches: 546.0 Pos Inicial: 1.0

BEST new error: 0.4458058435438266 Tamaño: 1061.0 Matches: 588.0 Pos Inicial: 2.0

BEST new error: 0.1969839773798303 Tamaño: 1061.0 Matches: 852.0 Pos Inicial: 3.0

5.0E-4 MegaIters

0.001 MegaIters

0.0015 MegaIters

0.002 MegaIters

0.0025 MegaIters

Óptimo: .1969839

Hay 33 estados en la Maquina de Turing

****

Complejidad de Kolmogorov: 528

****

EA	O	M	SE		O	M	SE
0	1	L	9		0	R	30
2	0	L	16		1	L	41
3	0	L	51		1	L	57
5	1	R	48		0	L	15
7	0	L	22		0	R	59

8| 0| R | 53|| 0| L | 4|

9| 1| R | 47|| 1| L | 41|

10| 1| R | 33|| 1| L | 5|

16| 1| R | 22|| 1| R | 7|

20| 1| R | 25|| 1| R | 17|

21| 1| R | 9|| 1| R | 2|

22| 0| R | 43|| 0| L | 39|

25| 1| R | 10|| 1| R | 2|

26| 0| L | 61|| 1| L | 45|

29| 0| L | 42|| 1| L | 44|

31| 1| L | 3|| 1| L | 7|

33| 0| R | 46|| 0| R | 54|

39| 0| R | 50|| 1| R | 50|

40| 1| L | 35|| 1| L | 39|

41| 0| R | 13|| 0| R | 49|

42| 0| R | 60|| 1| L | 34|

44| 0| L | 26|| 0| R | 48|

46| 1| L | 31|| 1| R | 46|

47| 1| L | 21|| 0| L | 54|

48| 1| L | 61|| 0| L | 54|

49| 1| L | 12|| 1| R | 25|

50| 0| L | 40|| 0| L | 29|

53| 1| R | 53|| 0| L | H|

54| 0| R | 3|| 0| R | 55|

55| 0| R | 20|| 0| L | 8|

57| 0| L | 0|| 0| L | 29|

60| 1| L | 5|| 0| R | 14|

61| 0| R | 9|| 0| L | 39|

Cadena esperada:
11001101011011101011111111111111111111111111111111110011010111001010110111111111111
11111111111111111111110101101011111111111111111111111111111111101101010111011110110
111001010101111111111111111111111111111111110111011111111111111111111111111111111111
11111010111110110101011011101011111101110001110011111111111111111111111111111111101
10101011101111010110101011101100101100110001110011111111111111111111111111111111111
01111011111111111111111111111111111111110101101011111111111111111111111111111110011
00010110110011111011101101011001001011000101111101111101010101011001010001110 10
1011011000111001111111111111111111111111111111111111101011111111111111111111111111111
1111110011000110110010100001100011111111111111111111111111111111110101110111011111
11111111111111111111111111110101100111011110001000011001010110101011111111111111111
11111111111111101011010111111111111111111111111111111111100110001011011001101011111
01001110010110110101101111111111111111111111111111111111110111101111111111111111111
111111111111010011000101011011011000100001100011011110001111 1

Cadena encontrada:
11001111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111

Prueba 5:

Cadena meta:

Wake up

Grab a brush and put a little (makeup)

Grab a brush and put a little

Hide the scars to fade away the (shakeup)

Hide the scars to fade away the

Why'd you leave the keys upon the table?

Here you go create another fable

You wanted to

Grab a brush and put a little makeup

You wanted to

Hide the scars to fade away the shakeup

You wanted to

Why'd you leave the keys upon the table?

You wanted to

I don't think you trust

In, my, self righteous suicide

I, cry, when angels deserve to die, die

Ejecución:

1) Funcion a optimizar: 33

2) Bits para enteros: 1023

3) Bits para decimales: 0

4) Numero de variables: 1

** Long. del genoma: 1024

5) Numero de iteraciones: 3000

6) Minimiza[0]/Maximiza[1]: 0

Modificar (S/N)? n

Deme el nombre del archivo de datos que quiere leer:

RESMT5.TXT

BEST new error: 0.9995917534190651 Tamaño: 4899.0 Matches: 2.0 Pos Inicial: 0.0

BEST new error: 0.1463563992651562 Tamaño: 4899.0 Matches: 4182.0 Pos Inicial: 0.0

BEST new error: 0.14594815268422123 Tamaño: 4899.0 Matches: 4184.0 Pos Inicial: 106.0

BEST new error: 0.14553990610328638 Tamaño: 4899.0 Matches: 4186.0 Pos Inicial: 9.0

5.0E-4 MegaIters

BEST new error: 0.1453357828128189 Tamaño: 4899.0 Matches: 4187.0 Pos Inicial: 2.0

0.001 MegaIters

0.0015 MegaIters

0.002 MegaIters

0.0025 MegaIters

Óptimo: .1453357

Hay 42 estados en la Maquina de Turing

****

Complejidad de Kolmogorov: 672

****

EA	O	M	SE	O	M	SE
0	0	L	46	1	L	22
1	1	R	13	1	L	27
3	0	L	33	0	L	35

4| 1| R | 40|| 1| L | 33|

6| 0| L | 62|| 0| L | 14|

7| 1| L | 36|| 1| L | 27|

8| 0| L | 18|| 1| R | 4|

10| 1| R | 24|| 1| L | 54|

12| 0| L | 57|| 0| R | 26|

13| 1| R | 25|| 0| R | 37|

16| 1| R | 26|| 1| L | 7|

17| 1| L | 58|| 0| R | 61|

19| 1| L | 44|| 0| L | 41|

21| 0| R | 40|| 0| L | 18|

24| 1| R | 54|| 0| R | 8|

25| 0| L | 16|| 0| L | 58|

26| 1| L | 60|| 0| L | 55|

27| 1| R | 61|| 1| R | 55|

28| 0| R | 58|| 1| L | 26|

31| 0| L | 32|| 0| L | 52|

32| 1| R | 1|| 0| L | 3|

33| 0| L | 1|| 1| R | 58|

34| 0| R | 16|| 0| L | 9|

35| 0| R | 17|| 1| R | 48|

36| 0| L | 16|| 0| L | 55|

37| 0| R | 37|| 1| L | 50|

39| 1| R | 45|| 0| L | 6|

40| 1| L | 10|| 1| R | 12|

41	1	R	11	0	L	46
44	1	R	12	1	L	34
45	1	L	13	0	R	62
46	0	L	39	0	L	7
48	0	R	21	1	R	4
50	0	R	28	1	R	17
52	0	R	32	0	L	56
54	1	R	38	1	L	31
55	1	R	55	1	L	24
57	1	L	7	1	L	35
58	1	R	13	1	R	50
60	0	R	57	0	R	57
61	1	L	39	1	R	16
62	0	L	6	1	L	19

Cadena esperada:
1000001010101001110111111111111111111111111111111111111011001100001101110101011111
11111111111111111111111111110101111111111111111111111111111111111011110111111011100 1
000111111111111111111111111111111111110101011111011111111111111111111111111111111111100
1111101110111111111111111111111111111111111010111111111111111111111111111111111111010110
01011101111011010111101111111111111111111111111111111111111111111111111111111111111111111
1011010101010011101111011001111111111111111111111111111111111110000110111010101111111
1111111111111111111111111110101111111111111111111111111111111111110111101111111011100100
011111111111111111111111111111111110101011111011111111111111111111111111111111111111100111
110111011111111111111111111111111111111101011111111111111111111111111111111111101011001
0111011110110101111010001100101101111011111111111111111111111111111111111110110001111
011111111111111111111111111111111111001100101011011111001111111111111111111111111111111
111111011100011111111111111111111111111111111111101011011110111111111111111111111111
1111111110101000000101010001011111111111111111111111111111111111111011000111101111111111
11111111111111111111111111111111111111111111111111111111111001000110101010011101111011
00111111111111111111111111111111111110001100101101111011111111111111111111111111111111
1110110001111011111111111111111111111111111111111111001100101011011111001111111111111111
111111111111111111111011100011111111111111111111111111111111111111111010110111101111111111
11111111111111111111110101000000101010001011111111111111111111111111111111111111110110001
111010000001000110001011111111111111111111111111111111111111101111111111111111111111111111
1111100010110001111011111111111111111111111111111111111101011110101011111111101111111111

1111111111111111111111110110001111011111111111111111111111111111111111010011101000101
1100111111111111111111111111111111111111111011001110001011111111111111111111111111111
1111110110001111011111111111111111111111111111111111111011010101110101111011111111111
1111111111111111111000111101101111101111111111111111111111111111111111000101100011111
0111111111111111111111111111111111110000110001111111111111111111111111111111111100110111
1101010111011110111111111111111111111111111111111101010111110001110110001111011011111
1111111111111111111111111111111111101010111010111101000101100011110111111111111111111
1111111111111111000001010101111110111101101111111111111111111111111111111111111011100
0100001101110101011111111111111111111111111111111111101011111111111111111111111111111
1110111101111110111001000111111111111111111111111111111111111010101111101111111111111
1111111111111111111001111101110111111111111111111111111111111111101011111111111111111
1111111111111111101011000101110111101101011110111111111111111111111111111111111011010
1010100111011110110011000101100011110111111111111111111111111111111111000001010101
1111101111011011111111111111111111111111111111111011100010001100101101111011111111
1111111111111111111111111011000111101111111111111111111111111111111111100110010101101
1111100111111111111111111111111111111111111011100011111111111111111111111111111111111
0101101111011111111111111111111111111111111101010000010101000101111111111111111111
1111111111111101100011110111111111111111111111111111111111111001000110101010011101111
1011001100010110001111011111111111111111111111111111111111000001010101111110111010
1111111111111111111111111111111111111011100010000010001100010111111111111111111111111
1111111111011111111111111111111111111111111000101100011110111111111111111111111111111
1111111101011110101011111111101111111111111111111111111111111111111011000111101111111111
1111111111111111111111010011101000101110011111111111111111111111111111111111111011001
1100010111111111111111111111111111111111111011000111101111111111111111111111111111111
1111110110101011101011110111111111111111111111111111111111000101100011110111111111111
1111111111111111111111000001010101111110111101101111111111111111111111111111111111110
1110001001011111111111111111111111111111111111011100010111111111111111111111111111111
1111111011111111111111111111111111111111111011000110010101111010011111111111111111111
1111111111111100010110001111011111111111111111111111111111111111111011011111011100111
0110010101111111111111111111111111111111111111111111111111111111111111111111111101101000
1011111111111111111111111111111111111111111111111111111111111111111111111111111001110101011111111
1111111111111111111111111111111011100101000010001111011110110001111011100111111111
1111111111111111111111111001111010010110010010110111101001011111111111111111111111
1111111111111111111111111111111111111111111001101110001011111111111111111111111111111
1111111111111111111111111111111111111111000001000111101011111111111111111111111111111
1101010111100001110101011110011111111111111111111111111111111111111011101110011011011
1111111101111111111111111111111111111111111011100011111111111111111111111111111111110
1100101110111111111111111111111111111111111111111111111111111111111111111111111011001011
10

Cadena encontrada:
1010011010111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111

Resultados

Para las pruebas, se decidió utilizar una cadena binaria y cuatro cadenas de caracteres de diferentes tamaños, dos de las cuales pertenecen a secciones de poemas, una es una frase arbitraria y la última es la primera parte de una canción en inglés.

Al realizar las pruebas, existieron varios factores que se necesitan mencionar sobre los resultados.

Lo primero que pudimos concluir es que para llegar a una aproximación mucho más exacta sería necesario aumentar los parámetros iniciales: el tamaño de la cinta inicial, el número máximo de iteraciones de la Máquina de Turing y el número máximo de iteraciones que permite el análisis de Kolmogorov. Esto fue muy fácil de comprobar, ya que solo bastaba correr el código un par de veces con la misma cinta meta, pero modificando los valores que le asignamos a las variables ya mencionadas. Rápidamente podíamos notar que nuestro error disminuye debido a que el número de aciertos, al finalizar la simulación, aumentaba.

Por otro lado, en cuanto a la longitud de nuestros parámetros, también notamos que mientras más larga sea la cadena que utilizamos como cinta objetivo, más difícil es replicarla a la perfección. Esto se debe a que nuestros parámetros están bastante limitados y esto dificulta el cálculo de la complejidad de Kolmogorov. Si bien esta función es incomputable, se necesitan mejores herramientas computacionales para llevar a cabo una mejor aproximación que la que se obtuvo en las pruebas. Creemos que con un mayor poder de cómputo y al asignarle más iteraciones a nuestro código, los resultados serían más acertados.

Lo segundo que se puede abstraer es que al trabajar con cadenas largas, las mejores aproximaciones fueron aquellas que tenían cadenas completas de unos. Esto no siempre es cierto, ya que depende mucho de la cadena meta, pero las cadenas elegidas para las pruebas resultaron tener más unos que ceros y esto afectó la forma en cómo trabajó la función. La función fue buscando las cadenas que replicaban más unos y fue corrigiendo su aproximación con la primera parte de la cadena resultante. Esto resultó en que fuera más fácil encontrar en nuestra cinta obtenida bits con valores de uno que deberían ser cero a bits con valores de cero que deberían ser uno. Creemos que este comportamiento se debe a que, al tener cadenas tan largas y ofrecerles tan pocas iteraciones, las cadenas que más se aproximan a la cinta objetivo son aquellas que tratan de replicar puros unos.

Podemos finalizar diciendo que, para lograr un análisis más completo, es necesario realizar más pruebas y aumentar las limitaciones físicas que tiene nuestro proyecto; sin embargo, para fines explicativos, las pruebas exponen cómo trabaja el algoritmo y cómo logra los resultados.