Actividad:

Integradora 1

Diego Mellado Oliveros A01655451
Iwalani Amador Piaga A01732251
Tonatiuh Reyes Huerta A01025459

Campus Santa Fe

Análisis y diseño de algoritmos avanzados

TC2038

Introducción a la situación problema:

Actualmente se transmite una gran cantidad de información por medio de series de bits, pero el hecho de intercomunicarse por medio de redes hace posible el interceptar las transmisiones y con ello, modificar las partes de envio, ocasionando que al enviarlas al destinatario la información cambie e incluso se puedan insertar scripts o pequeños programas que puedan crear problemas para el destinatario, controlando o infectando su equipo.

Descripción del funcionamiento:

Con el escenario anteriormente planteado, se ha creado una simulación a lo que podría ser la detección de códigos maliciosos que pueden estar interviniendo en el envío de datos de una transmisión.
Para ello el programa lee archivos de texto que son procesados por el mismo programa, tal como en un envío de datos real, donde aunque la información se envía hexadecimalmente el usuario receptor no se encarga de dar instrucciones al programa, solo recibe la información procesada.

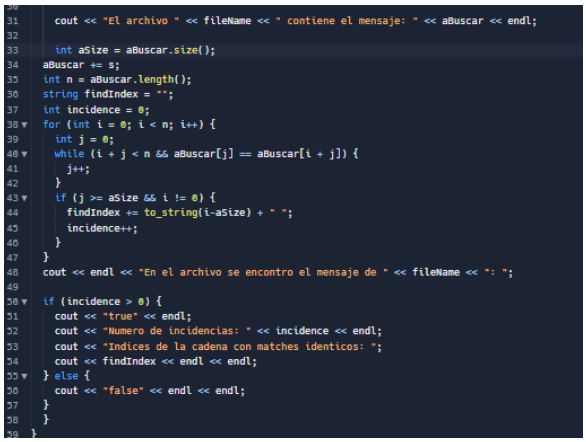
Lor archivos que en este caso lee el programa son, un archivo (“transmission.txt”) que representa un envío de datos de un equipo a otro, y 3 archivos (mcode)
representan código malicioso que se puede encontrar dentro de una transmisión después de que alguien malicioso intervino la comunicación. Donde la intención de analizar la transmisión y compararla con los encode para ver si están integrados, es para identificarlos y crear reportes por cada encode encontrado e informar cuantas veces se repite dentro de la transmisión y sus posiciones.

Algoritmos utilizados y su complejidad:

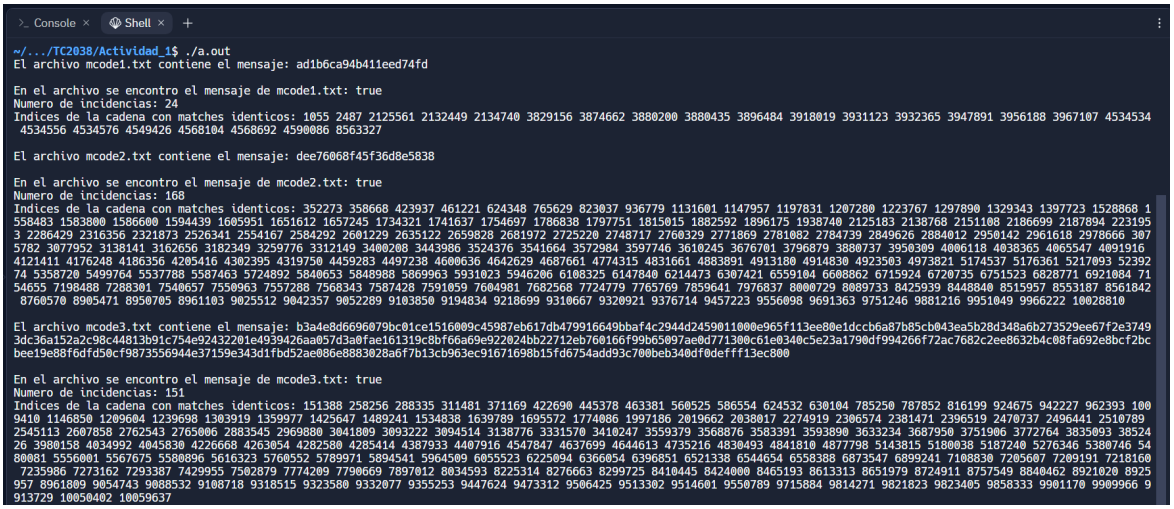
Para el desarrollo del programa se utilizó principalmente el algoritmo de Z-function para que comparara la transmisión con los archivos encode para encontrar las incidencias, que es el objetivo al hacer los reportes de la comunicación simulada, para llegar al resultado se complemento con arreglos KMP de incidencia y búsqueda para reportar sus índices en la cadena con matches idénticos a los mcode analizados. Al final se deduce que es O(n) por su tiempo de complejidad, donde la búsqueda de incidencias es también de complejidad O(n+m).
De esta forma, la solución implementada fue más dinámica a fin de lograr una menor complejidad posible sobre el coste de memoria utilizada por el programa.

Capturas del programa:

```cpp
#include <iostream>
#include <sstream>
#include <fstream>
#include <string>

using namespace std;

string readString(string fileName) {
  string s;
  ifstream inFile(fileName);

  string elem;

  while (getline(inFile, elem)) {
    s += elem;
  }

  inFile.close();
  return s;
}

int main() {
  string fileNames[3] = {"mcode1.txt", "mcode2.txt", "mcode3.txt"};
  string s = readString("transmission.txt");

  for (int i = 0; i < 3; i++) {
    string fileName = fileNames[i];
    string aBuscar = readString(fileName);
```



```cpp
    cout << "El archivo " << fileName << " contiene el mensaje: " << aBuscar << endl;

    int aSize = aBuscar.size();
  aBuscar += s;
  int n = aBuscar.length();
  string findIndex = "";
  int incidence = 0;
  for (int i = 0; i < n; i++) {
    int j = 0;
    while (i + j < n && aBuscar[j] == aBuscar[i + j]) {
      j++;
    }
    if (j >= aSize && i != 0) {
      findIndex += to_string(i-aSize) + " ";
      incidence++;
    }
  }
  cout << endl << "En el archivo se encontro el mensaje de " << fileName << ": ";

  if (incidence > 0) {
    cout << "true" << endl;
    cout << "Numero de incidencias: " << incidence << endl;
    cout << "Indices de la cadena con matches identicos: ";
    cout << findIndex << endl << endl;
  } else {
    cout << "false" << endl << endl;
  }
  }
}
```



```
Console × Shell × +

~/.../TC2038/Actividad_1$ ./a.out
El archivo mcode1.txt contiene el mensaje: ad1b6ca94b411eed74fd

En el archivo se encontro el mensaje de mcode1.txt: true
Numero de incidencias: 24
Indices de la cadena con matches identicos: 1055 2487 2125561 2132449 2134740 3829156 3874662 3880200 3880435 3896484 3918019 3931123 3932365 3947891 3956188 3967107 4534534
 4534556 4534576 4549426 4568104 4568692 4590086 8563327

El archivo mcode2.txt contiene el mensaje: dee76068f45f36d8e5838

En el archivo se encontro el mensaje de mcode2.txt: true
Numero de incidencias: 168
Indices de la cadena con matches identicos: 352273 358668 423937 461221 624348 765629 823037 936779 1131601 1147957 1197831 1207280 1223767 1297890 1329343 1397723 1528868 1
558483 1583800 1586600 1594439 1605951 1651612 1657245 1734321 1741637 1754697 1786838 1797751 1815015 1882592 1896175 1938740 2125183 2138768 2151108 2186699 2187894 223195
3 2286429 2316356 2321873 2526341 2554167 2584292 2601229 2635122 2659828 2681972 2725220 2748717 2760329 2771869 2781082 2784739 2849626 2884012 2950142 2961618 2978666 307
5782 3077952 3138141 3162656 3182349 3259776 3312149 3400208 3443986 3524376 3541664 3572984 3597746 3610245 3676701 3796879 3880737 3950309 4006118 4038365 4065547 4091916
4121411 4176248 4186356 4205416 4302395 4319750 4459283 4497238 4600636 4642629 4687661 4774315 4831661 4883891 4913180 4914830 4923503 4973821 5174537 5176361 5217093 52392
74 5358720 5499764 5537788 5587463 5724892 5840653 5848988 5869963 5931023 5946206 6108325 6147840 6214473 6307421 6559104 6608862 6715924 6720735 6751523 6828771 6921084 71
54655 7198488 7288301 7540657 7550963 7557288 7568343 7587428 7591059 7604981 7682568 7724779 7765769 7859641 7976837 8000729 8089733 8425939 8448840 8515957 8553187 8561842
 8760570 8905471 8950705 8961103 9025512 9042357 9052289 9103850 9194834 9218699 9310667 9320921 9376714 9457223 9556098 9691363 9751246 9881216 9951049 9966222 10028810

El archivo mcode3.txt contiene el mensaje: b3a4e8d6696079bc01ce1516009c45987eb617db479916649bbaf4c2944d2459011000e965f113ee80e1dccb6a87b85cb043ea5b28d348a6b273529ee67f2e3749
3dc36a152a2c98c44813b91c754e924322O1e4939426aa057d3a0fae161319c8bf66a69e922024bb22712eb760166f99b65097ae0d771300c61e0340c5e23a1790df994266f72ac7682c2ee8632b4c08fa692e8bcf2bc
bee19e88f6dfd50cf9873556944e37159e343d1fbd52ae086e8883028a6f7b13cb963ec91671698b15fd6754add93c700beb340df0defff13ec800

En el archivo se encontro el mensaje de mcode3.txt: true
Numero de incidencias: 151
Indices de la cadena con matches identicos: 151388 258256 288335 311481 371169 422690 445378 463381 560525 586554 624532 630104 785250 787852 816199 924675 942227 962393 100
9410 1146850 1209604 1239698 1303919 1359977 1425647 1489241 1534838 1639789 1695572 1774086 1997186 2019662 2038017 2274919 2306574 2381471 2396519 2470737 2496441 2510789
2545113 2607858 2762543 2765006 2883545 2969880 3041809 3093222 3094514 3138776 3331570 3410247 3559379 3568876 3583391 3593890 3633234 3687950 3751906 3772764 3835093 38524
26 3980158 4034992 4045830 4226668 4263054 4282580 4285414 4387933 4407916 4547847 4637699 4644613 4735216 4830493 4841810 4877798 5143815 5180038 5187240 5276346 5380746 54
80081 5556001 5567675 5580896 5616323 5760552 5789971 5894541 5964509 6055523 6225094 6366054 6396851 6521338 6544654 6558388 6873547 6899241 7108830 7205607 7209191 7218160
 7235986 7273162 7293387 7429955 7502879 7774209 7790669 7897012 8034593 8225314 8276663 8299725 8410445 8424000 8465193 8613313 8651979 8724911 8757549 8840462 8921020 8925
957 8961809 9054743 9088532 9108718 9318515 9323580 9332077 9355253 9447624 9473312 9506425 9513302 9514601 9550789 9715884 9814271 9821823 9823405 9858333 9901170 9909966 9
913729 10050402 10059637
```

Tres aplicaciones (distintas de la presentada en esta situación) de los algoritmos empleados en esta actividad.

- KMP: para la búsqueda de patrones, aplicable para reemplazar algoritmos hash. Una aplicación distinta al ámbito tecnológico, puede ser la investigación dentro del área de la medicina, ya que se buscan patrones a partir de datos obtenidos o registrados, se puede hacer una búsqueda de cadena de aminoácidos en las cadenas de proteína
- KPM: para la búsqueda de patrones, un programa de administración empresarial que analiza patrones de palabras prohibidas en el chat de una empresa, ya sea de forma interna o para valorar la experiencia que está teniendo un cliente a la hora de interactuar con un interno para la validación del buen servicio al cliente.
- Con Z function, podemos hacer una función de prefijo que a partir de esta se construya un autómata, calculando la función de prefijo para todos los caracteres posteriores sobre la marcha, para determinar la posición y los estados,

En otras palabras, podemos construir un autómata (una máquina de estados finitos): el estado en él es el valor actual de la función de prefijo, y la transición de un estado a otro se realizará a través del siguiente carácter.
Por lo tanto, incluso sin tener la cuerda t, podemos construir una tabla de transición de este tipo $(Old\pi, c) \rightarrow new$ usando el mismo algoritmo que para calcular una tabla de transición. Aumentando las posibilidades y el poder computacional a aplicaciones que pueden utilizar autómatas.

Otras aplicaciones de Z function:

- Comprimir una cadena de caracteres
- El número de subcadenas diferentes en una cadena
- Contar el número de apariciones de cada prefijo
- Buscar en la subcadena

Reflexión final:

Es importante conocer las implicaciones que pueden estar presentes en las intercomunicaciones creadas a partir de la globalización, pero esta actividad integradora no solo nos permitió recordar el funcionamiento de las transmisiones de información y su malversación, sino también, como las ciencias computacionales son utilizadas para cualquier infraestructura tecnológica y así, con el conocimiento adquirido en el curso es posible implementarlo en función a mejorar la vida de las personas actuando éticamente. Aprender a identificar códigos maliciosos es uno de los primeros pasos que podemos tomar para identificar vulnerabilidades que puedan ser explotadas.
Al final, el uso de “Z function” nos ayudó a identificar incidencias de códigos maliciosos, que forman parte de los archivos prueba que nos proporcionaron en archivos txt (mcode), el entender cómo funciona la lectura de strings junto con la aplicación de algoritmos avanzados nos ayudó a contextualizar la utilidad del algoritmo más allá de la situación problema, como los que se explican previamente antes de esta reflexión.
Finalmente como equipo y de forma individual esperamos poder implementar los algoritmos aprendidos en el curso en siguientes proyectos y actividades, tanto escolares como proyectos personales.