
PROYECTO SEÑAES IPC 2

202201211 – José Leonel López Ajvix

Resumen

Proyecto de la Facultad de Ingeniería creado con Python. Este proyecto trata de solucionar la problemática de los parámetros claves de las ondas de sonido agrupadas por frecuencia y amplitud. Para resolver este problema, se decidió usar una metodología de agrupamiento, en este proyecto se decidió crear tres matrices. Para esto se crea una matriz con todas sus señales ordenadas por amplitudes y tiempos. Luego se crea una segunda matriz que comparará todos los tiempos y amplitudes que tienen los datos, para luego poder crear una tercera matriz que sumará y agrupará todos los datos. Estos datos vendrán en un archivo XML el cual debe ser leído y ejecutado bajo ciertos parámetros. Para comodidad del usuario también se podrá crear un XML con la nueva matriz y se creará una gráfica para poder leer más cómodamente la matriz reducida. Además, el programa podrá inicializar los datos para que nuevos datos sean leídos.

Palabras clave

Señales, tiempo, amplitud, matriz, lista

Abstract

Project of the Faculty of Engineering created with Python. This project tries to solve the problem of the key parameters of sound waves grouped by frequency and amplitude. To solve this problem, it was decided to use a clustering methodology, in this project it was decided to create three matrices. For this, a matrix is created with all its signals ordered by amplitudes and times. Then a second matrix is created that will compare all the times and amplitudes that the data have, in order to then be able to create a third matrix that will add and group all the data. These data will come in an XML file which must be read and executed under certain parameters. For the convenience of the user, an XML can also be created with the new matrix and a graph will be created to be able to read the reduced matrix more comfortably. In addition, the program will be able to initialize the data so that new data is read.

Keywords

Signals, time, amplitude, matrix, list

Introducción

Este proyecto permitirá solventar la problemática del Centro de Investigación de la Facultad de Ingeniería, pudiendo crear una matriz reducida que agrupa los datos por tiempo y amplitud, además de contener las restricciones creadas, y poder sumar las que tengan coincidencias. Esto hará la lectura de las matrices más fácil y la facilidad de tener un proceso automatizado el cual la Facultad de Ingeniería no lo tenía automatizado. Para esto se usó varias listas simplemente enlazadas que podrán guardar estos datos y contenerlos de manera lógica, estos datos se almacenarán en la memoria volátil de la computadora, esto con la finalidad de poder guardar más datos si se es necesario y asimismo, tener la facilidad de no poseer una base de datos. Además, utilizando la extensión graphviz, se pudo crear una gráfica para que sea más fácil la lectura para personas que no comprenden cómo funciona un lenguaje de programación.

Desarrollo del tema

El desarrollo de esta aplicación, pudo automatizar una problemática que se tenía en la facultad de ingeniería. El centro de investigaciones de la Facultad de ingeniería decidió en que se creara tres matrices, una matriz donde se agrupará cada señal con su tiempo y amplitud, asociando el tiempo y amplitud de la siguiente manera.

TIEMPO (seg)	AMPLITUD (db)				
	1	2	3	4	
	1	2	3	0	4
	2	0	0	6	3
	3	3	4	0	2
	4	1	0	1	5
	5	0	0	3	1

Como se logra apreciar, los datos de la señal se logran reunir dependiendo del tiempo, de manera de tabla. Si bien, la lógica y presentación de la señal es de manera matricial, se puede trabajar de manera de una lista simplemente enlazada para poder facilitar operaciones, líneas de código y lógica en el proyecto.

Las restricciones de las señales son:

- Debe contener nombre, amplitud y tiempo
- El tiempo no podrá ser menor a 0 ni mayor o igual a 3600
- La amplitud deberá ser mayor a 0 y menor a 360

Del mismo modo, los datos de las señales tendrán las siguientes restricciones:

- Deberá contener el dato, la amplitud y el tiempo
- Si un tiempo o amplitud no están definidos en la entrada del archivo, tomará un valor predeterminado como 0
- El tiempo no puede ser mayor al tiempo de la señal contenida
- La amplitud no puede ser mayor a la amplitud de la señal contenida

Todos estos datos vendrán contenidos en un archivo XML, el cual contendrá la siguiente estructura:

```

1  <?xml version="1.0"?>
2  <senales>
3      <senal nombre="Prueba 1" t="120" A="100">
4          <dato t="1" A="1">2</dato>
5          <dato t="3" A="2">4</dato>
6          <dato t="1" A="4">4</dato>
7          <dato t="2" A="1">0</dato>
8          <dato t="3" A="4">2</dato>
9          <dato t="5" A="2">0</dato>
10         <dato t="2" A="2">0</dato>
11         <dato t="2" A="3">6</dato>
12         <dato t="2" A="4">3</dato>
13         <dato t="3" A="1">3</dato>
14         <dato t="3" A="3">0</dato>
15         <dato t="1" A="3">0</dato>
16         <dato t="4" A="1">1</dato>
17         <dato t="4" A="2">0</dato>
18         <dato t="5" A="3">3</dato>
19         <dato t="4" A="3">1</dato>
20         <dato t="4" A="4">5</dato>
21         <dato t="5" A="1">0</dato>
22         <dato t="1" A="2">3</dato>
23         <dato t="5" A="4">1</dato>
24         <dato t="6" A="1">1</dato>
25         <dato t="6" A="2">1</dato>
26         <dato t="6" A="3">0</dato>
27         <dato t="6" A="4">1</dato>
28     </senal>
29 </senales>
  
```

La segunda matriz, será una matriz de patrones, está deberá verificar si la matriz seleccionada contiene un dato o no. Si el dato en cierto tiempo y amplitud existe, se deberá escribir un 1, de caso contrario un 0, de la siguiente manera:

TIEMPO (seg)	AMPLITUD (db)				
		1	2	3	4
	1	1	1	0	1
	2	0	0	1	1
	3	1	1	0	1
	4	1	0	1	1
	5	0	0	1	1

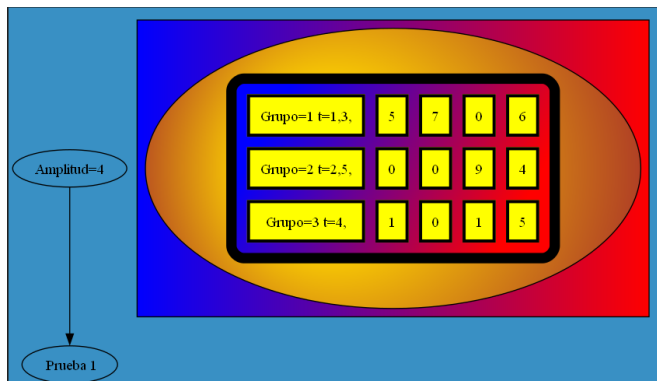
Esta matriz servirá para poder comprobar si existe un dato y asimismo poder agrupar los datos en la siguiente matriz de una manera más fácil. Si bien, la manera proporcionada es por medio de una matriz, se puede realizar de una lista simplemente enlazada para poder facilitar el código y la lógica del mismo.

Luego de que existan estas dos matrices, nos será más fácil calcular la lista patrones, la lista patrones vendrá ordenada de la siguiente manera.

TIEMPO (seg)	AMPLITUD (db)				
		1	2	3	4
	Grupo 1 (tiempo 1 y 3)	5	7	0	6
	Grupo 2 (tiempo 2 y 5)	0	0	9	4
	Grupo 3 (tiempo 4)	1	0	1	5

Como vemos, esta última matriz agrupa dependiendo del patrón dado en la matriz de patrones y además, suma los datos dependiendo del patrón, su amplitud y tiempo.

Para que la lectura sea mucho más fácil, podemos imprimir la señal deseada y nos generará una matriz como la mostrada anteriormente



También, para que la lectura sea mucho más fácil, se creará un documento XML, conteniendo la matriz reducida que contendrá el siguiente orden:

```
<?xml version='1.0' encoding='utf-8'?>
<Senales-reducidas>
  <senal nombre_archivo="Prueba 1" A="4">
    <grupo grupo="1">
      <tiempos>1,3,</tiempos>
      <datosGrupo>
        <dato A="1">5</dato>
        <dato A="2">7</dato>
        <dato A="3">0</dato>
        <dato A="4">6</dato>
      </datosGrupo>
    </grupo>
    <grupo grupo="2">
      <tiempos>2,5,</tiempos>
      <datosGrupo>
        <dato A="1">0</dato>
        <dato A="2">0</dato>
        <dato A="3">9</dato>
        <dato A="4">4</dato>
      </datosGrupo>
    </grupo>
    <grupo grupo="3">
      <tiempos>4,</tiempos>
      <datosGrupo>
        <dato A="1">1</dato>
        <dato A="2">0</dato>
        <dato A="3">1</dato>
        <dato A="4">5</dato>
      </datosGrupo>
    </grupo>
  </senal>
</Senales-reducidas>
```

Si nos damos cuenta, tanto el XML como la gráfica generada tiene el mismo orden y la misma lógica que la matriz generada anteriormente como ejemplo.

Al momento de iniciar la aplicación, existirá un menú inicial el cual será mostrado de esta manera:

```
Menu principal:
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo de salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Inicializar sistema
7. Salida
```

El cual mostrará las opciones como se muestran

Si elegimos la opción 1, nos mostrará las siguientes opciones:

```
1

Escriba el nombre del archivo:
```

Donde escribiremos el nombre y extensión del archivo. Luego al procesar el archivo, empezará a procesar todos los datos y también a crear la matriz reducida:

```
2

=====
senal: Prueba 1 tiempo: 5 amplitud: 4
datos:
dato: 2 tiempo: 1 amplitud: 1 senal Prueba 1 binario 1
dato: 3 tiempo: 1 amplitud: 2 senal Prueba 1 binario 1
dato: 0 tiempo: 1 amplitud: 3 senal Prueba 1 binario 0
dato: 4 tiempo: 1 amplitud: 4 senal Prueba 1 binario 1
dato: 0 tiempo: 2 amplitud: 1 senal Prueba 1 binario 0
dato: 0 tiempo: 2 amplitud: 2 senal Prueba 1 binario 0
dato: 6 tiempo: 2 amplitud: 3 senal Prueba 1 binario 1
dato: 3 tiempo: 2 amplitud: 4 senal Prueba 1 binario 1
dato: 3 tiempo: 3 amplitud: 1 senal Prueba 1 binario 1
dato: 4 tiempo: 3 amplitud: 2 senal Prueba 1 binario 1
dato: 0 tiempo: 3 amplitud: 3 senal Prueba 1 binario 0
dato: 2 tiempo: 3 amplitud: 4 senal Prueba 1 binario 1
dato: 1 tiempo: 4 amplitud: 1 senal Prueba 1 binario 1
dato: 0 tiempo: 4 amplitud: 2 senal Prueba 1 binario 0
dato: 1 tiempo: 4 amplitud: 3 senal Prueba 1 binario 1
dato: 5 tiempo: 4 amplitud: 4 senal Prueba 1 binario 1
dato: 0 tiempo: 5 amplitud: 1 senal Prueba 1 binario 0
dato: 0 tiempo: 5 amplitud: 2 senal Prueba 1 binario 0
dato: 3 tiempo: 5 amplitud: 3 senal Prueba 1 binario 1
dato: 1 tiempo: 5 amplitud: 4 senal Prueba 1 binario 1
=====
```

Luego de que se crean todas las señales y datos, podremos escoger la opción 4 para crear el XML de la lista reducida:

```
Escriba el nombre y la extensión del archivo
ejemplo
```

Nos creará el XML.

Podremos ver los datos del estudiante con la opción 4:

```
Los datos del estudiante son:
> José Leonel López Ajvíx
> 202201211
> Introducción a la programación y computación sección D
> Ingeniería en Ciencias y Sistemas
> 4to Semestre
```

Con la opción 5, podremos crear las matrices de las diferentes señales de nuestro archivo.xml

```
Escriba el nombre de la señal que desea imprimir gráfica
Prueba 1

Seleccione 1 si desea graficar la matriz normal
Seleccione 2 si desea graficar la matriz binaria
Seleccione 3 si desea graficar la matriz reducida
1
```

Y dependiendo de las gráficas, nos creará la gráfica de la matriz que deseamos graficar.

La opción 6 nos dejará inicializar el sistema, y así poder procesar nuevos archivos con nuevas señales

```
6

Programa inicializado con éxito!!
```

Y por último, podremos cerrar el programa:

```
7

Feliz día :)
```

Como logramos ver, la aplicación cumple las necesidades de manera óptima y sin ningún error en el flujo. Esta aplicación puede resolver y automatizar la problemática planteada en el centro de Investigaciones en el Área de la Facultad de Ingeniería.

Conclusiones

- Este programa logrará automatizar la problemática de la compresión de señales de audio.
- Con el uso de la aplicación Python y listas simplemente enlazadas, se logró crear una aplicación que resuelva todas las

problemáticas sin ningún problema o singularidad al momento de la ejecución.

- Con la implementación de esta aplicación, se logró crear un archivo XML de lectura y poder generar gráficas, esto servirá para que personas con pocos conocimientos de programación también puedan entender el flujo y lógica de la aplicación y como se procesan las señales.

Referencias bibliográficas

- González Duque, R. (2011). Python para todos.
- Van Rossum, G., & Drake Jr, F. L. (2017). *El tutorial de Python*. Python Software Foundation.
- Romero Munizaga, J. L. (2022). Librería de visualización de estructuras de datos.

Anexos

Diagrama de clases:

