

Algoritmos y Estructuras de Datos I

Primer cuatrimestre de 2020

Versión 1 (4 de mayo de 2020)

TPE - Reuniones Remotas

Fecha de entrega: 22 de Mayo (hasta las 17hs)



1. INTRODUCCIÓN

En los tiempos que corren atravesamos la necesidad de conectarnos remotamente con otras personas, ya sea individual o grupalmente. En este contexto usamos programas que nos permiten interactuar con nuestros interlocutores. Nos interesa resolver una serie de problemas relacionados con el habla, que se dan cuando un grupo de personas mantiene una conversación online. La voz de los interlocutores será captada a través de sus micrófonos y almacenadas en secuencias de datos con las cuales vamos a trabajar.

2. DEFINICIONES

Las ondas sonoras son variaciones continuas en la presión del aire, esas variaciones de presión se convierten a través de un micrófono en una señal eléctrica. Existen dos maneras de almacenar esas señales, de forma analógica o digital.

En este trabajo utilizaremos audios almacenados de manera digital representados con una secuencia de valores enteros a los que llamaremos muestra.

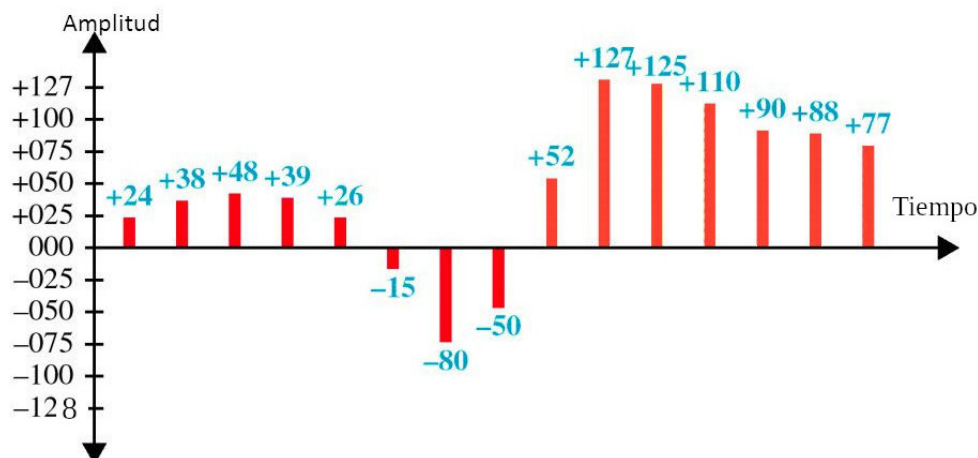
Cada muestra representa la amplitud de la señal en un punto específico en el tiempo. El rango de las muestras quedará determinado por el número de bits de información en cada una de ellas (profundidad). Siendo P la *profundidad*, el rango queda definido de la siguiente manera:

$$[-2^{(P-1)}, 2^{(P-1)} - 1]$$

Los Cd's utilizan P igual a 16, es decir, 16 bits por muestra, mientras que para el Blu-ray, P es 24.

Por último, la *frecuencia de muestreo* definirá la cantidad de veces que nuestro equipo recoge una muestra de la señal por segundo, 1000 muestras por segundo equivalen a 1 kHz. Algunos ejemplos: Teléfono (voz) 8 kHz, Radiodifusión 32 kHz, CD 44.1 kHz. A partir de ahora cuando hablemos de frecuencia estará medida en kHz.

En el siguiente ejemplo vemos una señal analógica codificada para muestras digitales de 8 bits; la profundidad de bits es ocho, por lo que la amplitud de cada muestra es uno de los 256 valores posibles dentro del rango $[-2^{(8-1)}, 2^{(8-1)} - 1] = [-128, 127]$



Además contaremos con reuniones, una reunión es una secuencia de tuplas $señal \times hablante$ que representa la señal que tiene cada persona en una conversación. Dentro de una reunión, todas las conversaciones comienzan y terminan al mismo tiempo (tienen la misma cantidad de muestras). Además, no puede haber más de una señal para la misma persona. Las personas estarán representadas con números enteros empezando en 0 (en una reunión de n personas, se les asignarán los números en el rango $[0..n - 1]$). Por ejemplo, si en una reunión participan 3 personas, las personas hablando se podrían representar con la secuencia $[1, 0, 2]$.

3. EJERCICIOS

Dados el siguiente renombre de tipos:

```
type señal = seq<ℤ>
type hablante = ℤ
type reunion = seq<señal × hablante>
```

Especificar los siguientes problemas:

Ejercicio 1. `proc esSeñal(in s: seq<ℤ>, in prof : ℤ, in freq: ℤ, out result : Bool)`: dada una secuencia s , comprueba si es una señal. Diremos que una secuencia es una señal cuando la frecuencia de muestreo es 8Khz o 32 Khz, la profundidad 8, 16 o 32 bits y dura más de 1 segundo.

Ejercicio 2. `proc seEnojó?(in s: señal, in umbral: ℤ, in prof : ℤ, in freq: ℤ, out result: Bool)`: dada la señal de un hablante indicará si este se enojó. Diremos que un hablante está enojado si a lo largo de la conversación la amplitud supera el umbral por más de 5 segundos consecutivos.

Ejercicio 3. `proc esReuniónVálida(in r: reunion, in prof : ℤ, in freq: ℤ, out result : Bool)`: dada una secuencia r , comprueba si es una reunión válida.

Ejercicio 4. `proc acelerar(inout r: reunion, in prof: \mathbb{Z} , in freq: \mathbb{Z})`: dada una reunión, acelera el hablar de cada uno de los participantes. Una señal será acelerada cuando posea la mitad de las muestras de la original, tomando solamente las muestras que se encuentran en posiciones impares.

Ejercicio 5. `proc ralentizar(inout r: reunion, in prof: \mathbb{Z} , in freq: \mathbb{Z})`: dada una reunión, lentificar el hablar de cada uno de los participantes. Una señal será lentificada cuando posea el doble de las muestras originales. Se crearan muestras ficticias entre medio de cada par de puntos de la señal, las mismas serán el promedio de los dos puntos vecinos.

Ejercicio 6. `proc tonosDeVozElevados(in r: reunion, in freq: \mathbb{Z} , in prof : \mathbb{Z} , out hablantes: seq(hablante))`: Dada una reunión, determina cuales fueron los hablantes que tienen el tono de voz más elevado, siendo el tono de voz el promedio de las amplitudes de la señal.

Ejercicio 7. `proc ordenar(inout r: reunion, in freq: \mathbb{Z} , in prof : \mathbb{Z})`: Que reordena las señales de la reunión según el tono de voz del hablante (de mayor a menor).

Ejercicio 8. `proc silencios(in s: señal, in prof: \mathbb{Z} , in freq: \mathbb{Z} , in umbral: \mathbb{Z} , out intervalos: seq(intervalo))`: Que dada una señal, determina los silencios (pares de indices de inicio y final del silencio). Diremos que hay un silencio si en cierto momento de la señal, todo valor absoluto de la misma no pasa cierto umbral por al menos 0.1 segundos. Tomar en cuenta que antes del inicio y después del final de los silencios encontrados, no puede seguir habiendo silencio y que los intervalos no pueden estar repetidos.

Ejercicio 9. `proc hablantesSuperpuestos(in r: reunion, in prof: \mathbb{Z} , in freq: \mathbb{Z} , in umbral: \mathbb{Z} , out result: Bool)`: Que dice si en algún momento hay más de una persona hablando (tomando en cuenta la definición de silencio del punto anterior).

Ejercicio 10. `proc reconstruir(in s: señal, in prof: \mathbb{Z} , in freq: \mathbb{Z} , out result: señal)`: Dada una señal, si posee elementos faltantes, es decir, su valor es cero, la señal debe ser reconstruida de la siguiente manera: si en una posición hay un valor faltante, se debe completar con el promedio de los dos valores no nulos de la secuencia original más cercanos.

Términos y condiciones

El trabajo práctico debe realizarse de manera grupal y todos los integrantes del grupo deben conocer como se resuelven los ejercicios. Para aprobar el trabajo se necesita:

- Que todos los ejercicios estén resueltos.
- Que las soluciones sean correctas.
- Que el lenguaje de especificación esté bien utilizado.
- Que las soluciones sean prolijas: evitar repetir especificaciones innecesariamente y usar adecuadamente las funciones y predicados auxiliares.
- Que no haya casos de sub-especificación ni sobre-especificación.

Pautas de Entrega

El trabajo debe ser subido al campus en la sección Trabajos Prácticos en la fecha estipulada.

Fecha de entrega: 22 de Mayo (hasta las 17hs)