
Tareas de SPSI

curso 2022/2023

fco. m. garcía olmedo

17 de octubre de 2022

Índice

1. Normas	1
2. Tareas (40 % del 50 %)	2

1. Normas

1. Para su realización, el alumno usará en cada ejercicio el lenguaje sugerido por considerar que es el idóneo para su ejecución.
2. La entrega de las tareas será vía la plataforma seleccionada al efecto y en un único fichero que resulte de comprimir una carpeta estructurada por tareas con todo lo que el alumno desee entregar de entre las las propuestas en este fichero que está leyendo.
3. El formato de compresión será .zip o tar.gz y no otro. En caso de usar otro formato de compresión no ofrecemos garantía de que sea descomprimido para su revisión y subsiguiente evaluación.
4. El nombre del fichero entregado tendrá uno de los siguientes formatos (y no otro):
 - spsi_2021_2022_tareas_inicialesAlumno_dniAlumno.zip
 - spsi_2021_2022_tareas_inicialesAlumno_dniAlumno.tar.gz
5. Cualquiera de las tareas que entregue en alumno deberá incluir al menos una versión completa de la implementación en el formato de Jupyter, que es .ipynb, con independencia de que quiera ofrecer otra en formato .py —lo que puede ser interesante y hasta conveniente.

6. El contenido de la implementación debe ser suficientemente comentado y ello contará como una parte de la nota.
7. La copia de código supondrá la invalidación de la práctica, con sus correspondientes consecuencias en la media.
8. Estas tareas podrán ser hechas en el grupo de prácticas que ha sido consolidado para el trabajo en laboratorio, pero cada alumno deberá hacer entrega independiente y obligatoria del trabajo en común para recibir puntuación. Todos los alumnos del grupo harán entrega de exactamente el mismo fichero a excepción del nombre del mismo, que contendrá rasgos identificativos según lo dicho.
9. Una parte de la nota será obtenida por la fiel observancia de lo expresado en los puntos anteriores.

2. Tareas (40 % del 50 %)

Tarea 1) En los primeros días de la computación fueron construidos algunos ordenadores soviéticos bajo la aritmética ternaria equilibrada. El más famoso fue el *Setun*, construido por *Nikolay Brusentsov* y *Sergei Sobolev*. La mencionada aritmética tiene varias ventajas sobre la aritmética binaria. Particularmente, la consistencia más-menos reduce la proporción de acarreo en multiplicaciones con números multidígito y el truncamiento con redondeo reduce la proporción de acarreo al redondear fracciones. Además de otras muchas ventajas, la representación ternaria equilibrada permite representar los enteros sin el uso del signo menos; el valor del dígito líder (no nulo) tiene el signo del número mismo.

La sucesión finita de números enteros $e_0e_1 \cdots e_{n-1}e_n$ es una expresión ternaria equilibrada del número entero b si, por definición, cumple las siguientes condiciones:

- a) Para todo $0 \leq i \leq n$, $e_i \in \{-1, 0, 1\}$
- b) $b = \sum_{i=0}^n e_i 3^i$
- c) $e_n \neq 0$

El hecho de que se cumplan estas condiciones es abreviado escribiendo:

$$b = (e_n e_{n-1} \cdots e_1 e_0)_{te}$$

Para todo número entero no nulo b existe una única representación ternaria equilibrada. Elabore un programa en Python presentado en al menos un fichero .ipynb que calcule la expresión ternaria equilibrada de cualquier número entero.

Tarea 2) Implemente en Python (o Haskell) un simulador de la máquina de [Hagelin modelo C-52](#) y haga que su implementación sea configurable con el estilo mostrado para el caso de la máquina Enigma en [@ringstellung](#), ofreciendo un diálogo similar al de los ejemplos de este sitio.