

Instrucciones de Entrega

La solución a este taller debe subirse por SICUA antes de las 8:30AM del jueves 27 de Agosto del 2015. Si la solución está en SICUA antes de las 8:30AM del sábado 22 de Agosto del 2015 se calificará el taller sobre 130 puntos. Los tres archivos código fuente deben subirse en un único archivo `.zip` con el nombre `NombreApellido_hw2.zip`, por ejemplo yo debería subir el zip `JaimeForero_hw2.zip`. Todas los algoritmos deben ser implementados con funciones básicas de python. Librerías externas solamente son permitidas al momento de escribir/leer datos o hacer gráficas.

1. 30 (40) pt **Barajando Arreglos** En este punto van a implementar un algoritmo para desordenar arreglos.

- (a) 20 (25) pt Escriba una función que desordene un arreglo de n elementos de manera aleatoria (puede pensar en una baraja de cartas, por ejemplo). La función debe poder utilizarse desde python de la siguiente manera:

```
import barajando as baraja
import numpy as np
a = np.arange(10) #este es un array inicial de 10 elementos
b = baraja.baraja(a) #b es el array barajado
```

- (b) 10 (15) pt Escriba una función para visualizar la efectividad de este algoritmo. La función debe generar una matriz A_{ij} donde el elemento ij corresponde al número de veces que el elemento que inicialmente se encuentra en la posición i termina en la posición j si se hicieran muchas iteraciones desordenando el array. La función debe poder utilizarse desde python de la siguiente manera.

```
import barajando as baraja
matriz_a = baraja.matriz(baraja.baraja, n_lado_matriz=100, n_iter=10000)
```

2. 30 (40) pt **Volando sobre una esfera**

Los aviones que vuelan largas distancias internacionales buscan minimizar su consumo de combustible. Para esto la ruta de vuelo debe ser la más corta entre los puntos de partida y llegada. El objetivo de este ejercicio es escribir un programa que calcule las coordenadas de esa trayectoria para dos puntos arbitrarios sobre la Tierra.

Para simplificar el problema supondremos que la Tierra es perfectamente esférica. Además vamos a utilizar los ángulos θ (polar) y φ (ecuatorial) para describir las coordenadas sobre un punto de la esfera .

Escriba una función que dados dos puntos arbitrarios sobre la esfera (parametrizados por θ y φ) calcula la trayectoria más corta que entre esos dos puntos. La función debe poder utilizarse de la siguiente manera

```
import esfera as sf
```

```

theta_inicial = 0 # en grados
phi_inicial = 30 # en grados
theta_final = 240
phi_final = 10
n_points = 1000 # numero de puntos para la trayectoria
camino_theta, camino_phi = sf.camino(theta_inicial, phi_inicial,
theta_final phi_final, n_points)

```

3. 40 (50) pt **Cambio de coordenadas** Como vemos cada noche las estrellas realizan un movimiento aparente en el cielo, en realidad este movimiento se debe a la rotación de la tierra sobre su propio eje y alrededor del Sol. En el catalogo `hipparcos.csv`¹ se encuentra la siguiente información `id AR DEC Mag Distance` de cada estrella, Donde `AR` y `DEC` son las coordenadas ecuatoriales² de las estrellas. `Mag` Esta relacionado con la luminosidad de la estrella y `Distance` es la distancia a la estrella en parsecs³.

Escriba una función que transforme las coordendas del sistema ecuatorial (`AR` y `DEC`) a las coordenadas horizontales de `Altura` y `Azimut`. La función se debe poder utilizar sobre los datos de `Hipparcos` y llamarse desde python de la siguiente manera

```

import coordenadas as coor
filename = 'hipparcos.csv'
latitud = +45.0 # en grados
longitud = -34.0 # en grados
mes = 10
dia = 14
anno = 2014
hora = 20
minutos = 10
segundos = 30
hipp_ra, hipp_dec = coor.load_data(filename)
hipp_alt, hipp_az = coor.ec2hor(hipp_ra, hipp_dec,
lat=latitud, long=longitud, y=anno, m=mes, d=dia, h=hora, min=minutos,
s=segundos)

```

¹<https://github.com/ComputoCienciasUniandes/HerramientasComputacionalesDatos/tree/master/data/Hipparcos>

²http://es.wikipedia.org/wiki/Coordenadas_ecuatoriales

³<http://es.wikipedia.org/wiki/Parsec>