

Programación—Certamen 2 - Jueves 12 de Mayo de 2016

Nombre: Rol: -

2. [40 %] Una organización internacional de conservación construyó un domo de cristal para mantener un micro clima en una cierta superficie. Para esto, se instalaron miles de paneles de cristal, contando cada uno de ellos con varios sensores que reportan información diariamente a un computador central. Cada sensor transmite una vez al día una tupla con su medición, indicando la fecha, su identificador único y el valor de la medición. Como ejemplo, una medición sería: (2016, 3, 27, 'aa:01:10', 0.0021).

Todas estas mediciones son colocadas en una lista en el computador central, la cual es mantenida **ordenada descendentemente** por el valor medido. Esto facilita la recuperación posterior de mediciones, debido a que por el gran volumen del registro resulta imposible ordenarlo con cualquiera de los algoritmos disponibles en la actualidad. A modo de **ejemplo**, un posible registro sería:

```
registro = [  
  (2016, 1, 25, 'd2:00:10', 0.0112),  
  (2015, 10, 24, '3e:15:c2', 0.0105),  
  (2015, 11, 3, '28:0b:5c', 0.0009), # ...  
]
```

Usted debe:

- a) Construir la función `agregar_medicion(registro, A, M, D, id, val)`, la que agrega la información capturada al registro. Considere A el año, M el mes y D el día de la medición.

```
>>> agregar_medicion(registro, 2016, 3, 27, 'aa:01:10', 0.0021)  
>>> print registro  
[(2016, 1, 25, 'd2:00:10', 0.0112), (2015, 10, 24, '3e:15:c2', 0.0105),  
  (2016, 3, 27, 'aa:01:10', 0.0021), (2015, 11, 3, '28:0b:5c',  
  0.0009)]
```

- b) Construir la función `corregir_medicion(registro, A, M, D, id, nvo_valor)` que corrige el valor medido de un sensor. La función recibe como parámetro el registro, el año, mes y día de la medición a corregir, el identificador único y el nuevo valor medido. Considere que luego de esta actualización el registro debe permanecer ordenado como se indicó en el enunciado.

```
>>> corregir_medicion(registro, 2016, 3, 27, 'aa:01:10', 0.0221)  
>>> print registro  
[(2016, 3, 27, 'aa:01:10', 0.0221), (2016, 1, 25, 'd2:00:10', 0.0112),  
  (2015, 10, 24, '3e:15:c2', 0.0105), (2015, 11, 3, '28:0b:5c',  
  0.0009)]
```

- c) Construir la función `sensores_sobre_umbral(registro, u)`, la que recibe un valor de medición usado como umbral y también el registro. Esta función entrega un listado con todos aquellos sensores, cuyo valor medido se encontró por sobre el umbral u entregado. Este listado no debe contener identificadores duplicados.

```
>>> sensores_sobre_umbral(registro, 0.01)  
['aa:01:10', 'd2:00:10', '3e:15:c2']
```

Programación—Certamen 2 - Jueves 12 de Mayo de 2016

Nombre: Rol: -

3. [40 %] Una empresa de encomiendas almacena la información de despacho de valijas haciendo uso de las siguientes estructuras:

- `encomiendas` corresponde a un diccionario, donde la llave representa el código de una valija y su valor corresponde a una tupla, con el largo, alto y ancho de ésta en centímetros.
- `destinos` es un conjunto de tuplas. Cada tupla está compuesta por el código de la valija y la tupla (x,y) correspondiente al punto donde debe ser entregada en *km*.

A continuación se presenta un **ejemplo** de las estructuras:

```
encomiendas = {                                destinos = set([
    10234: (90, 90, 50),                        (10234, (-4, 15)),
    34102: (100, 140, 20),                      (36890, (2, 4)),
    36890: (25, 50, 70),                        (34102, (18, -13)),
    # ...                                       # ...
})
```

El ingreso (monetario) que obtiene la empresa por trasladar una valija depende del volumen (cm^3) y la distancia. La fórmula para calcular el ingreso en pesos es:

$$ingreso = 0,1 \cdot volumen \cdot distancia$$

Donde el volumen se obtiene multiplicando largo, alto y ancho de la valija (cm^3) y la distancia se calcula como $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Considere que la empresa está ubicada en el punto $(0,0)$ y los camiones se mueven en línea recta desde la empresa al punto de destino a una velocidad de $60km/h$.

La empresa dispone de 10 camiones y se desea, de ser posible, asignarle a cada uno un último despacho en el día, el cual no debe tardar más de 30 minutos en llegar a destino. Por supuesto se desea hacer tal asignación buscando el mayor ingreso para la empresa.

Para ello le solicita a usted crear la función `viajes_a_realizar(encomiendas, destinos)` la cual recibe como parámetro el diccionario `encomiendas` y el conjunto `destinos` y retorna una lista con a lo más 10 códigos correspondientes a las valijas a despachar que reportarán el mayor ingreso a la empresa.

Usted puede crear las funciones que estime necesarias para la resolución de este problema.