

Programación—Certamen 2 - Martes 7 de Noviembre de 2017

Nombre

Rol — Paralelo

2. [40 %] El exitoso juego Pukamon GOES entra en etapa de actualización, para implementar batallas entre entrenadores que poseen diversos Pukamon. Para ésto se utilizará la información contenida en las siguientes estructuras: El diccionario `pukadex` contiene como llave el nombre del Pukamon, y como valor una tupla con el tipo principal, el tipo secundario y el número del pukamon (el tipo secundario no siempre está presente). El diccionario `relacion` contiene como llave el tipo de un Pukamon, y como valor una tupla con todos los tipos de Pukamon que son débiles respecto a la llave.

```
pukadex = {
  'Pukachu1S': ('Electrico', 'Psiquico', '026'),
  'Bulmasaur': ('Planta', 'Veneno', '001'),
  'Kaku Topo': ('Electrico', 'Hada', '785'),
  'NewTwo': ('Psiquico', '', '150'),
  'Pukachu': ('Electrico', '', '025'),
  'Escuartul': ('Agua', '', '007'),
  'Lukagri': ('Pelea', 'Metal', '448'),
  'Chorizard': ('Fuego', 'Volador', '006'),
  'Donphunk': ('Tierra', '', '232'), #...}

relacion = { 'Dragon': ('Dragon'),
  'Fuego': ('Planta', 'Hielo', 'Bicho', 'Metal'),
  'Electrico': ('Agua', 'Volador'),
  'Volador': ('Planta', 'Pelea', 'Bicho'),
  'Agua': ('Fuego', 'Tierra', 'Roca'), #... }
```

- a) Desarrollar la función `pukamon_tipo(pukadex, tipo)` que retorna una lista de tuplas con todos los Pukamon del tipo `tipo` ordenados ascendentemente por número de Pukamon. **Debe respetar el formato y los tipos de datos del ejemplo.**

```
>>> pukamon_tipo(pukadex, 'Electrico')
[(25, 'Pukachu'), (26, 'Pukachu1S'), (785, 'Kaku Topo')]
```

- b) Desarrollar la función `tipos_desventaja(tipo1, tipo2, relacion)`, la cual retorna un conjunto de tipos de Pukamon que son débiles respecto del `tipo1` o `tipo2` de **un** Pukamon. Nota: `tipo1` y `tipo2` seran siempre strings no vacíos definidos en el diccionario `relacion`

```
>>> tipos_desventaja('Fuego', 'Volador', relacion)
set(['Hielo', 'Metal', 'Planta', 'Bicho', 'Pelea'])
```

- c) Desarrollar la función `pukamon_desventaja(pukadex, relacion, pukamon)` cuyo valor de retorno es una lista de todos los pokemones débiles respecto al Pukamon `pukamon`. La lista debe estar ordenada por el número del Pukamon.

```
>>> pukamon_desventaja(pukadex, relacion, 'Escuartul')
[(6, 'Chorizard'), (232, 'Donphunk')]
```

Programación—Certamen 2 - Martes 7 de Noviembre de 2017

Nombre

Rol

Paralelo

3. [40%] La entidad Py Quake Alert ha creado un sistema de predicción de terremotos el cual utiliza la información histórica de eventos sísmicos que ocurren en el mundo. El registro de los eventos sísmicos se guarda en una lista llamada `eventos`, la que contiene tuplas con la siguiente información: (`dia`, `mes`, `anno`, `hora`, `pais_de_ocurrencia`, `ciudad`, `magnitud`). A continuación, se muestra un ejemplo de la estructura `eventos`:

```
eventos = [ (19, 9, 2017, '10:01', "Mexico", "Ciudad de Mexico", 7.3),
            (17, 8, 2017, '11:31', "Mexico", "Ciudad de Mexico", 2.4),
            (15, 8, 2017, '04:07', "Chile", "Coquimbo", 6.3),
            (29, 7, 2017, '20:01', "Haiti", "Puerto Principe", 3.2),
            (11, 6, 2017, '12:22', "Chile", "Coyhaique", 5.3), ...]
```

En base a lo anterior, se le solicita construir la función `generar_predicciones(eventos, fecha)`, la que recibe la lista de eventos sísmicos `eventos` y una fecha en una tupla con el `dia`, `mes`, `anno`. Esta función debe retornar los siguientes 3 sismos a ocurrir considerando una ventana de tiempo de **6 meses** hacia atrás, desde la fecha entregada como parámetro. Para estos efectos considere:

- Todos los meses **tienen 30 días**.
- Se deben buscar los **3 países** con mayor cantidad de sismos en la ventana de tiempo.
- La intensidad del sismo vaticinado corresponderá **al promedio de los sismos** registrados en la ventana de tiempo para el respectivo país.

Notar que la primera variable a considerar para escoger un país es la cantidad de sismos. Sin embargo, si hay empates en cantidad de sismos, habiendo más de 3 países candidatos, se debe optar por aquel país cuya intensidad promedio de sismos sea mayor para generar la lista de 3 predicciones. Si aún así empatan, tomar cualquiera de los empatados para generar la lista de 3.

La función debe retornar una lista con 3 tuplas con el formato (`país`, `intensidad`). En caso de existir menos de 3 tuplas en el rango de tiempo solicitado, se retorna las que existan. Ver ejemplos para comprender lo que se espera como solución en cada caso:

```
>>> generar_predicciones(eventos, (20,10,2017))
[('Mexico', 4.85), ('Chile', 5.8), ('Haiti', 3.2)]
>>> generar_predicciones(eventos, (18,3,2018))
[('Mexico', 7.3)]
>>> generar_predicciones(eventos, (1,1,2578))
[]
```