

Ejercicios Dicionarios y Conjuntos

Miguel Solis

<http://www.innovacionyrobotica.com/~miguel.solis>

Facultad de Ingeniería - Escuela de Informática

Universidad Andrés Bello

Pregunta 1

Considere el sitio de citas online *MatchUnab.com*, sitio que aún se encuentra en etapa de construcción, por lo que le han pedido ayuda a los alumnos de Programación para completarlo y poder ayudar a tanto compañero solitario.

El diccionario *usuarios* contiene el nombre de cierta persona, y un conjunto que contiene sus cualidades, en el siguiente formato:

`usuarios = { 'Tom Cruise' : { 'simpatico' , 'comprensivo' } , 'Scarlett Johansson' : { 'buena onda' , 'atletica' } , 'Brad Pitt' : { 'comprensivo' } }`

- a) Implemente la función *buscar(cualidades , usuarios)*, que reciba el diccionario usuarios anteriormente descrito, de forma que retorne una lista con el nombre de todas las personas que tengan las cualidades nombradas en el conjunto cualidades.

Ejemplo:

```
cualidades = { 'comprensivo' }  
buscar( cualidades , usuarios )  
[ 'Tom Cruise' , 'Brad Pitt' ]
```

- b) Implemente la función *compatible(persona , cualidades , usuarios)*, que reciba el nombre de alguna persona, y un conjunto con sus cualidades, y retorne una tupla con el nombre de la persona del diccionario usuarios que tenga una mayor cantidad de cualidades en común, y el porcentaje de cualidades que tienen en común.

Ejemplo:

```
cualidades = { 'atletica' }  
compatible(cualidades , usuarios )  
( 'Scarlett Johansson' , 50.0 )
```

Pregunta 2

Con el boom de las aplicaciones online que requieren de una conexión a internet para funcionar correctamente, se hace necesaria la dependencia de acceso a la red. Agobiado de este tipo de vida, cierto profesor ha decidido tomarse unas vacaciones en una zona rural muy alejada, en la que sólo se cuenta con los servicios básicos (internet no está dentro de ellos).

Como tampoco es la intención de esta persona distanciarse afectivamente de sus amistades, le ha encargado a sus alumnos que hagan un programa que permita verificar las fechas de cumpleaños de sus amigos, que se encuentran almacenadas en el diccionario *fechas* en el siguiente formato:

```
fechas = {'Camilo': (05,10,1984), 'Roberto': (15,08,1983), 'Valentina':  
(18,04,1992), 'Nadia': (09,04,1991)}
```

- a) Implemente la función *cumple(m, fechas)* que retorna una lista de tuplas correspondientes a todos los nombres y días en que se encuentra de cumpleaños cada persona en el mes *m*.

Ejemplo:

```
cumple(4,fechas)  
[('Valentina', 18), ('Nadia', 9)]
```

- b) Implemente la función *edades(f, fechas)* que retorne un diccionario correspondientes a todos los nombres dentro del diccionario *fechas*, con la edad correspondiente calculada para la fecha *f*, donde *f* se encuentra en el formato (dia,mes,año).

Ejemplo:

```
hoy = (15,05,2014)  
edades(hoy,fechas)  
{ 'Camilo': 29, 'Roberto': 30, 'Valentina': 22, 'Nadia': 23}
```

Pregunta 3

La alimentación es un aspecto importante a considerar en el diario vivir, por lo que para ayudar a los clientes a alimentarse de forma más responsable cuidando lo que uno ingiere, el local de comida rápida Progaritico le ha pedido ayuda para realizar el siguiente programa.

Considere el diccionario *comidas* que contiene el índice calórico de cierto plato, en el siguiente formato:

`comidas = {'completo': 300, 'churrasco': 400, 'papas fritas': 200, 'especialidad': 1500}`

- a) Implemente la función *livianos($n, comidas$)* que retorne un conjunto de n tuplas, con los n platos más livianos (menor índice calórico).

Ejemplo:

```
livianos(2,comidas)
{('papas fritas',200), ('completo',300)}
```

- b) Implemente la función *regular($tope, comidas$)* que **modifique** el diccionario (y retorne el diccionario modificado) *comidas*, eliminando todos los platos que tengan un índice calórico mayor a *tope*.

Ejemplo:

```
regular(350,comidas)
{'completo': 300, 'papas fritas': 200 }
```

Pregunta 4

Los precios de los productos de un supermercado están almacenados en un diccionario como el siguiente:

```
precios = { 'aceite' : 500 , 'arroz' : 2000 , 'cafe' : 1500 , 'te' : 300 }
```

- a) Escriba la función *producto_mas_caro(precios)*, que reciba como parámetro el diccionario de precios, y retorne el nombre del producto más caro. Si hay más de un producto más caro, retorne cualquiera de ellos.

Ejemplo:

```
producto_mas_caro(precios)
'arroz'
```

- b) Una compra puede ser representada como un diccionario en que se asocia a cada producto la cantidad de unidades que se comprará.

Escriba la función *monto_total(precios, compra)*, que reciba como parámetro el diccionario de precios y el diccionario de la compra, y retorne el monto total a pagar por la compra.

Ejemplo:

```
monto_total(precios, { 'arroz': 2, 'cafe': 1 })
5500
```

Pregunta 5

Considere las clasificatorias al nacional de ciclismo *Pedaleando por la Programación*, donde participan los 2 mejores ciclistas de cada región de Chile. Se considera que un ciclista es mejor que otro si su tiempo récord es menor.

Considere que los tiempos récord (en minutos) de cada ciclista junto a sus respectivos nombres, están almacenados en un diccionario *tiempos* con el siguiente formato (considere que los nombres no se repiten):

```
tiempos = { # region: [('Nombre1',Tiempo1), ..., ('Nombren',Tiempon)],
             'V': [('Juan',39), ('Roberto',30),('Pedro',38)],
             'RM': [('Mario',40),('Jose',35),('Marcelo',45),('Manuel',42)],
             #...}
```

- a) Implemente la función *clasificados(region,tiempos)* que retorna un conjunto con los nombres de los ciclistas clasificados de dicha *region*.

Ejemplo:

```
clasificados('V',tiempos)
{'Roberto','Pedro'}
```

- b) Implemente la función *clasifica(ciclista,tiempos)* que retorna **True** si dicho *ciclista* ha clasificado al nacional o **False** en caso contrario.

Ejemplo:

```
clasifica('Marcelo',tiempos)
False
```

- c) Considere que despues de ejecutarse la competencia nacional, los resultados con el tiempo obtenido por cada ciclista clasificado se encuentran en la lista resultados con el siguiente formato:

```
resultados = [('Roberto',35), ('Pedro',34), ('Mario',36), ('Jose',33), ...]
```

Implemente la función *ranking(tiempos,resultados)* que retorne un diccionario cuya llave sea cada región que participó, y sus valores sean una lista con tuplas conteniendo el nombre de los ciclistas competidores y su posición lograda en el nacional.

Ejemplo:

```
ranking(tiempos,resultados)
{'V': [('Pedro',2),('Roberto',3)], 'RM': [('Mario',4),('Jose',1)]}
```