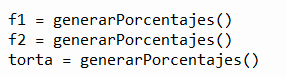
Corte supremo

Introduccion:

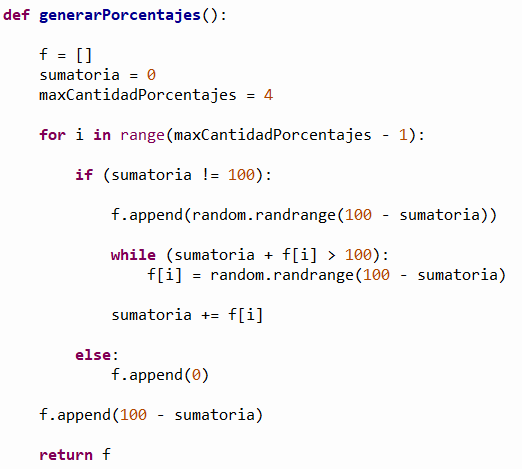
Dado el problema en el que teniendo los gustos de dos personas (u1 y u2) sobre los 4 sabores de una torta, se pedía que se aplicara una metodología de “Cut and choose” en la cual un usuario cortaba la torta en un parte y el otro usuario elegía con que parte quedarse, dejándole lo restante al usuario que haya cortado.

Para eso se pidio que se diseñara un programa el cual pudiera obtener una forma optima para que el usuario corte la torta en forma beneficiosa para el, sabiendo cuales eran los gustos de el usuario que eligiria con que parte quedarse. Pudiendo repetir este mismo proceso para una cantidad de N veces.

Inicio:

 Antes de comenzar con la resolucion de este problema, habria que primero poder definir mis datos a usar, es decir, los gustos de los usuarios y las proporciones entre los gustos de la torta.

Para eso, en mi caso aplique una misma función para ello, denominada “generarPorcentajes”

 El funcionamiento se basa en generar 4 valores, dando el primero para el porcentaje del primer gusto de la torta y así en orden hasta llegar al cuarto.

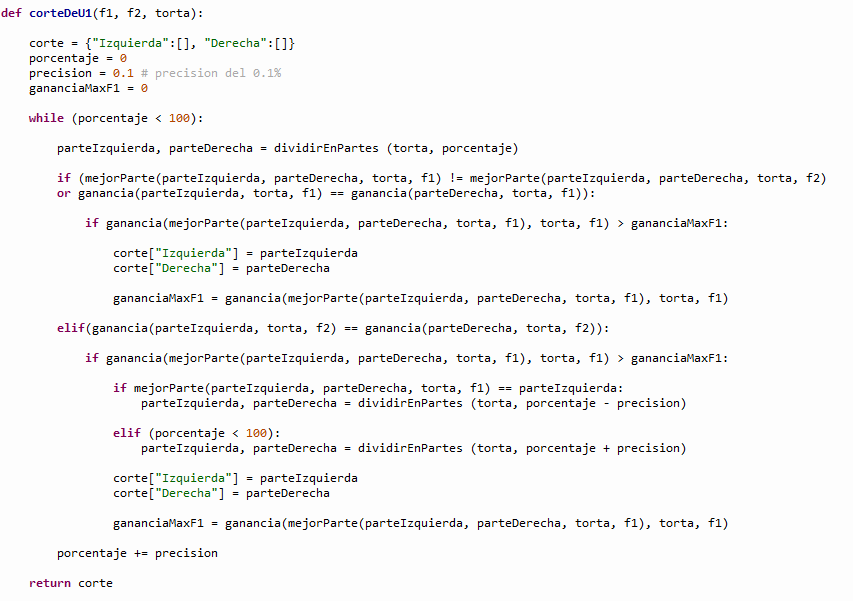
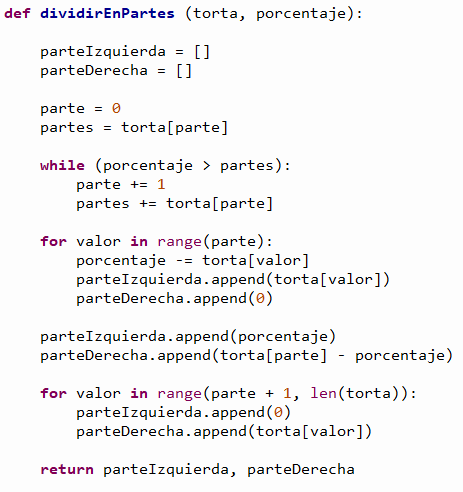
Para el primer valor no es mucho problema ya que se genera un numero random entre 0 y 100, ya para el segundo se necesita acortar el rango, si es que el primer valor, no fue 100 se generaría un valor entre 0 y (100 - porcentaje del primero) y lo mismo para el tercero, pero agregando el valor del segundo también, teniendo en cuenta que la sumatoria de los 2 o 3 valores no puede superar el 100, y de ser que el primer valor generado sea 100 o el segundo valor generado + el primero den 100, entonces se reemplazara el valor del segundo y/o tercero a 0; ya por ultimo el cuarto valor sera el valor restante, ya se 0 o mayor.

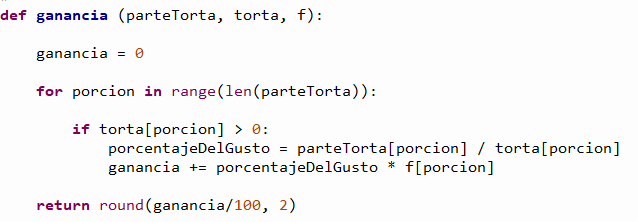
Con los distintos porcentajes ya declarados en las variables (f1: gustos del usuario que corta, f2: gustos del usuario que elige y torta: porcentajes de cada gusto en la torta) se tienen que cumplir 2 procesos, el hecho que el primer usuario tiene que cortar y el segundo debe de elegir.

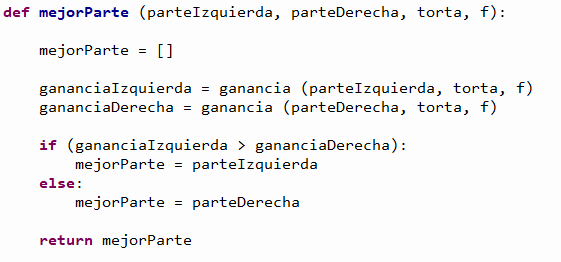
Para ello aplico las siguientes funciones, de la siguiente manera ...CS_3

**CorteDeU1:**

Ya como el nombre lo indica esta función se encarga de buscar cual es el corte donde el usuario que corta pueda verse mas beneficiado sabiendo que cuales son los gustos del otro usuario. Para eso aplique el siguiente algoritmo...

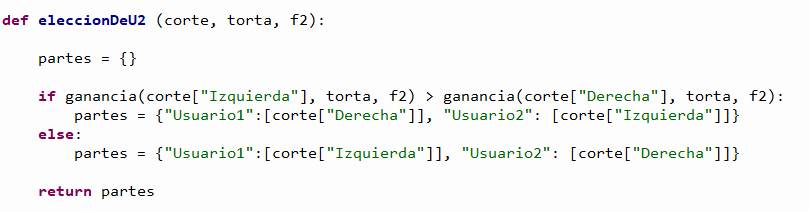
 En palabras simples, esta función va moviéndose de a 0.1% de la torta hasta llegar a completarla en un 100%, cada vez que esta avanza divide a la torta en dos partes, la parte izquierda y la parte derecha con la función dividirEnPartes, la cual, dada una torta y un porcentaje, divide a la torta en dos parte siendo que la parte izquierda se igual al porcentaje indicado y el derecho el resto. Una vez dividida analiza, si la parte que tendría mas ganancia para el usuario que corta es diferente es diferente que la del usuario que elige (osea usuario que corta: parte izquierda, usuario que elige: parte derecha), o si es que la ganancia de el lado izquierdo y el lado derecho son las mismas para el usuario que corta, ambas con la ayuda de las funciones denominadas **mejorParte** y **ganancia ...**

****



De ser que alguno de esos casos se cumpla, y la ganancia sea mayor a la mas grande hasta ahora (la cual en un principio es 0) se guardaran las partes en los debidos lugares dentro del diccionario corte, se actualizara el valor máximo de ganancia hasta entonces y se seguirá analizando el siguiente valor hasta llegar al 100%. De ser algun caso en las cuales estas dos condiciones no se cumplan, pero si que la ganancia para la parte izquierda y la parte derecha sean las mismas para el usuario que elige, entonces se analizara si es que la posible ganancia de alguno de los lados es mayor a la máxima hasta el momento para el usuario que corta y entonces, dependiendo cual sea el lado con mayor beneficio para el usuario que corta, este cortara un valor de precisión (en este caso 0.1%) para ese lado, dándole las porcentaje al lado que no le benéfica, incitando al otro usuario a elegir ese lado.

eleccionDeU2:

 Por utlimo, una vez el primer usuario decidió como cortar la torta el usuario que elige, analiza las dos partes cortadas de la torta dadas por **CorteDeU1** y elige aquella que le da una mayor ganancia.

Para concluir podemos decir que este tipo de división es justa y libre de envidia ya que ambos siempre consiguen la mejor parte posible de las partes de la torta, y, por ultimo el proceso se repite de la siguiente manera, utilizando todas las funciones mencionadas hasta ahora...

