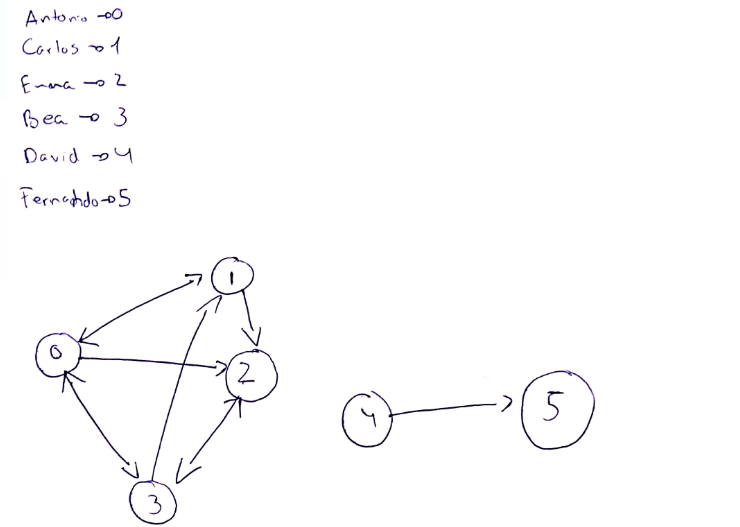
# PROBLEMA 1

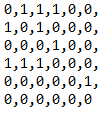
Javier Martín Gómez

Lenguaje utilizado: Python

En este problema, el objetivo es averiguar cuál es el grado de conexión de una red de usuarios. Para representar esta red vamos a usar una matriz inicial (lista de listas) que simule al grafo donde 1 marca conexión con el nodo y 0 marca que no hay conexión. Por ejemplo, el grafo del ejemplo sería el siguiente:



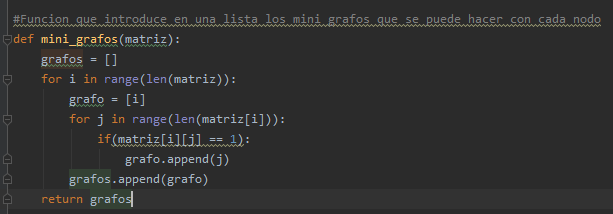
Como vemos hay dos grupos (dos grafos). Ahora vemos cómo sería la matriz:



Ahora, tenemos que realizar un algoritmo voraz para, a partir de la matriz, hallar cuántos grafos (grupos) se pueden sacar de la matriz. Para ello, primero implementamos una función que devuelva una lista con las conexiones de cada nodo incluyendo el nodo. Por ejemplo, en el caso del ejemplo, devolverá lo siguiente:

[[0, 1, 2, 3], [1, 0, 2], [2, 3], [3, 0, 1, 2], [4, 5], [5]]

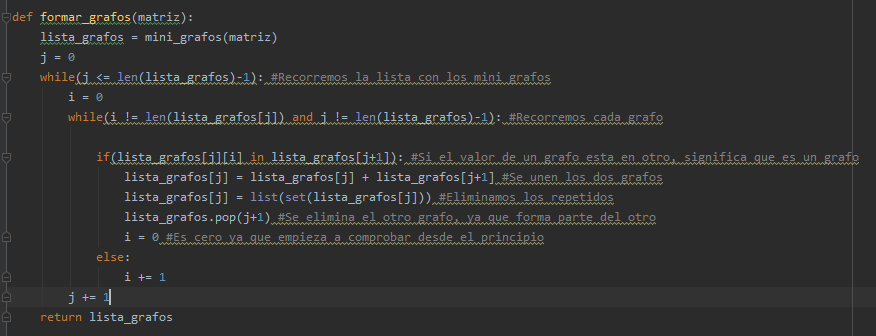
Donde la primera sublista son las conexiones del primer nodo, la segunda sublista del segundo nodo, y así sucesivamente. El código es el siguiente:



Ahora, partiendo de la lista con las conexiones de cada nodo, tenemos que hallar cuántos grafos se pueden formar con dichas conexiones. Para ello, empezamos recorriendo la lista formada por las conexiones de los nodos. Recorremos las conexiones de los nodos para comprobar si están en otra sublista, si lo están, ambos nodos estarán unidos, por lo que se concatenan las dos listas. Por ejemplo, para el primer nodo ([0, 1, 2, 3]) empezamos cogiendo el 0 y comprobamos si está en el segundo([1, 0, 2]). Al estar, se concatenan las listas, pasa a ser una lista única formada por [0,1,2,3] (con set se eliminan las repeticiones) y la segunda lista se elimina. Se va haciendo sucesivamente hasta recorrer toda la lista. La longitud de la lista será el número de grupos. En nuestro caso sería esta:

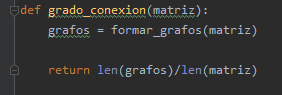
[[0, 1, 2, 3], [4, 5]]

Vemos que hay 2 grupos (grafos), como nos había salido mecánicamente. El código es el siguiente:



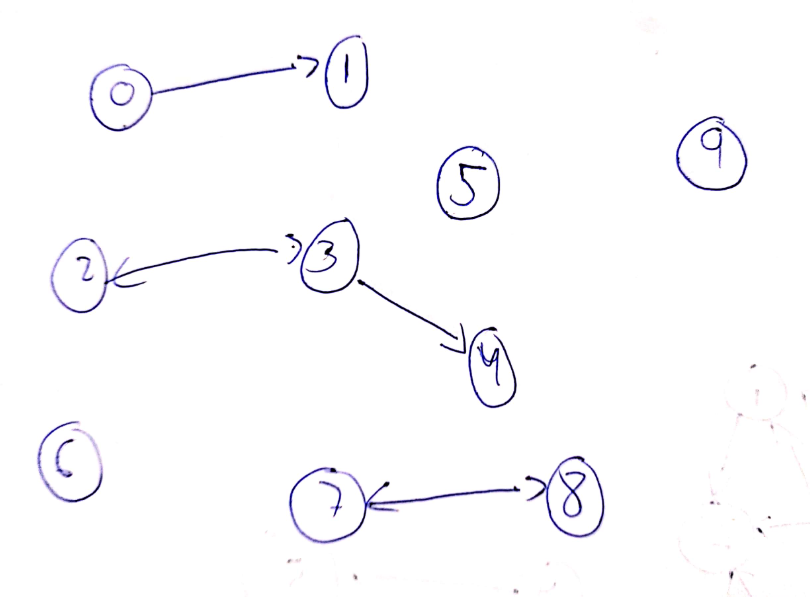
Para terminar, hallamos el grado de conexión, que divide la cantidad de grupos (número de grafos o longitud de la lista anterior) entre el número de usuarios (longitud de la matriz).

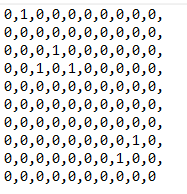
En nuestro caso, sería 2/6=0.3333

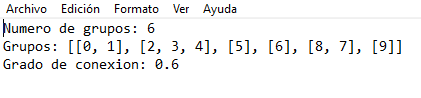


Más ejemplos:

ejemplo\_voraz2.txt







ejemplo\_voraz3.txt

