Primera parte: IMDB

En este TP vamos a analizar los datos de IMDB (https://www.imdb.com/). Actualmente, IMDB es la base de datos más grande del mundo donde se encuentran programas de televisión, eventos en vivo y difundidos en televisión o en la web, entrega de premios, especiales y videojuegos, con más de 100 millones de usuarios activos. Los datos que deberán poner en la carpeta "datasets" se encuentran acá y son un subset con solo películas y actores.

Para realizar este análisis, les vamos a proveer un grafo (A) que modela las distintas relaciones que existen entre actores y películas y ustedes deberán construir el grafo B.

A) El primer grafo no dirigido contiene a los distintos artistas en los vértices, y las aristas van a conectar a dos artistas si y sólo si colaboraron juntos en alguna película. Además, las aristas van a contener un set con todos los títulos de película en las que estos colaboran.
B) El segundo grafo no dirigido va a contener tanto a las películas como a los artistas en sus vértices. En este caso, una película y un artista van a estar conectados mediante una arista si y sólo si dicho artista fue parte del elenco de esa película. En este caso, las aristas no tienen un peso asociado. Como curiosidad, este es un ejemplo de un grafo bipartito, donde los nodos se pueden particionar en películas y artistas respectivamente. https://es.wikipedia.org/wiki/Grafo_bipartito

Ejercicios

1) **[Grafo A]** Nos interesa entender cuan "particionado" esta nuestro grafo. Con esta información podríamos "clusterizar" a nuestra comunidad de artistas. Se pide hallar la cantidad de componentes conexas que lo componen, y asignar a cada vértice en una misma componente conexa un mismo identificador. Una componente conexa se define como un subgrafo maximal conexo. ¿Cuántas componentes conexas hay? ¿Cuál es la segunda componente conexa más grande? ¿Cúal es la más chica de todas?
2) **[Grafo B]** Dados dos actores, queremos conocer su grado de separación. El grado de separación se define como la mínima cantidad de películas de distancia a la que se encuentran.
3) **[Grafo B]** Encuentre quien está a mayor grado de separación de Kevin Bacon en su componente conexa
4) **[Grafo A]** Dado un artista, queremos conocer el camino mínimo (que sume la minima cantidad de colaboraciones totales) a todos los demás artistas.
5) **[Grafo A]** Queremos conocer el camino **minimo** (con pesos) entre **cualquier** par de artistas. ¿Cuánto demora calcular esto?
6) **[Grafo A]** ¿Cuál es el diámetro del grafo para la componente conexa principal? Utilice máximo 15 minutos para calcularlo, si no le es posible calcularlo de forma exacta, indique:
    a) ¿Cuánto demoraría hacerlo de forma exacta?
    b) Estime el diámetro del grafo utilizando el tiempo dado
7) **[Grafo A]** ¿Cuál es, en la componente conexa principal, el promedio de separaciones para cada actor y para todos en general? De ser imposible de calcular estimelo.
8) **[Grafo B]** Por medio de random walks estime quienes son los vértices con mayor centralidad, diferenciando actores de películas. (+1 puntos)

9) **[Grafo A]** ¿Quienes son los actores con más *betweenness centrality*? (+1 puntos)
10) **[Grafo A]** El coeficiente de clustering de un grafo correlaciona fuertemente en redes sociales con depresión**.** ¿Quiénes son los actores que haciendo más de 3 películas tienen menor coeficiente de clustering? (+2 puntos)

Segunda parte: Sudoku

En *sudoku.py* hay implementado un *grafo implícito* sobre el sudoku. Implemente un algoritmo basado en grafos que sea capaz de resolver cualquier sudoku.