

Grados de separación entre actores.

Partimos de un grafo de datos sobre actores y películas. Según la teoría de los [grados de separación de Kevin Bacon](#) cualquier persona de la industria del cine de Hollywood puede conectarse con Kevin Bacon en un máximo de 6 pasos donde cada paso es una película en la que han participado dos personas. En este problema os pedimos encontrar el camino más corto entre dos actores dados, a través de la secuencia de películas que los conectan. Por ejemplo, el camino más corto entre Jennifer Lawrence y Tom Hanks tiene longitud 2: Jennifer Lawrence se conecta con Kevin Bacon por la película "X-Men: First Class," y Kevin Bacon con Tom Hanks por "Apollo 13."

Podemos definir este problema como un problema de búsqueda en el que los estados son las personas y las acciones son las películas que nos llevan de un actor al siguiente<sup>1</sup>. El estado inicial y final son las personas que queremos conectar. Resuelve el problema indicando qué algoritmo es el más adecuado.

Se os proporciona la base de dato en formato csv con actores que participan en películas de dos tamaños (pequeño y grande en dos directorios) y el código Python para cargar los datos a través de estructuras diccionario. Se puede consultar una persona por su nombre (y obtener su id) y las películas en las que ha participado. La función `load_data` carga los archivos csv en los diccionarios.

En el fichero de personas (`people.csv`) cada persona tiene un id único<sup>2</sup>, un nombre y su año de nacimiento (102,"Kevin Bacon",1958). Y en el archivo de películas (`movies.csv`) cada película tiene también un id único, su título y el año (109830,"Forrest Gump",1994). En el tercer archivo (`stars.csv`) se establece la relación entre las personas y las películas. La persona con id 102 participó en la película con id 104257.

En primer lugar, debes descargar y entender el contenido de los archivos y estructuras de datos y el código que os damos ya hecho para leer los datos, obtener el id dado el nombre y obtener las conexiones de una persona dada. Dado el id de una persona, la función `neighbors_for_person` devuelve el conjunto de pares (`movie_id`, `person_id`) con todas las personas que participaron con esa persona en la película con `movie_id`. Si hay un camino entre `pi` y `po`, el camino entre ellas puede ser devuelto como una lista de pares de enteros (`movie_id`, `person_id`). Por ejemplo, si el camino más corto es: [(1, 2), (3, 4)], significa que `pi` participó en la película 1 con la persona 2, la persona 2 participó en la película 3 con la persona 4. Y como termina el camino, 4 es `po`.

Ejemplo de salida:

---

<sup>1</sup> Aunque en el paradigma del espacio de estados normalmente las acciones deben ser deterministas, observar que en este problema una misma película puede conectar una persona con varias.

<sup>2</sup> Que corresponde con su identificador en [IMDb](#)

Loading data...

Name: Emma Watson

Name: Jennifer Lawrence

3 grados de separación

1: Emma Watson y Brendan Gleeson participaron en Harry Potter and the Order of the Phoenix

2: Brendan Gleeson y Michael Fassbender participaron en Trespass Against Us

3: Michael Fassbender y Jennifer Lawrence participaron en X-Men: First Class

Si hay varios caminos de la misma longitud se puede devolver cualquiera o puedes proponer algún criterio adicional (por ejemplo, usando el año). Puede ocurrir que no haya camino entre dos actores.