

Movies

Hugo Gómez Sabucedo

14 de noviembre de 2024

Introducción

En este proyecto se desarrolla en Python un análisis básico de datos sobre películas de cine de IMDB. El set de datos que vamos a usar inicialmente se encuentra en la siguiente página:

<https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset>

En ella puede encontrarse información más detallada, así como una descripción precisa de cada columna.

Se plantean los apartados iniciales para su resolución *sin usar la librería pandas*. Esto se propone así para forzar a practicar con las técnicas, herramientas y conceptos básicos de Python. Más adelante, hay otros apartados propuestos para su resolución con esta librería y otras.

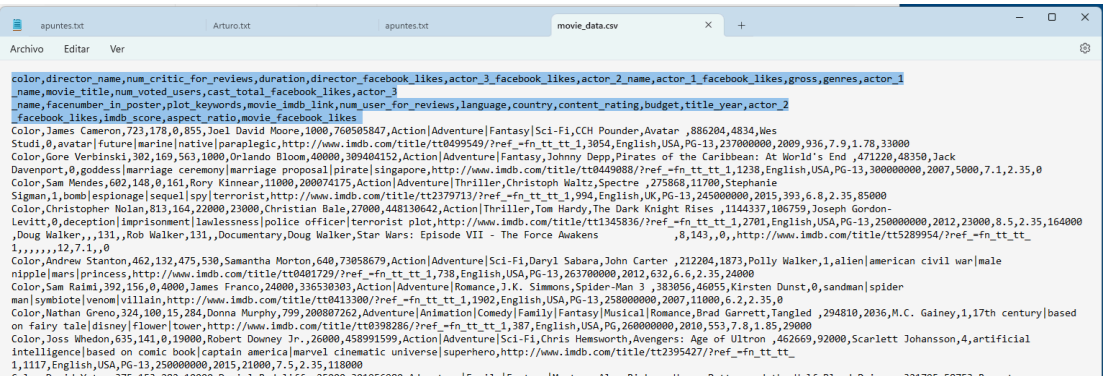
A partir de los datos proporcionados, he conseguido realizar correctamente todos los apartados propuestos.

Aunque al final de este notebook detallaré la calificación que calculo honestamente, globalmente, siguiendo las puntuaciones que se asigna a cada apartado, diría que he obtenido una nota de 9 sobre 10.

Datos de partida

(**Nota previa:** hasta el apartado específico de dataframes, se deben desarrollar las soluciones a los ejercicios propuestos sin usar la librería `pandas`, justamente para obligar a practicar con conceptos básicos de Python.)

Nuestra tabla de datos es el archivo de texto `movie_data.csv` que se encuentra en la carpeta `data_in`, y puede verse así con cualquier editor:



La primera fila es la cabecera. Esta fila cabecera contiene los nombres de los campos, separados por comas. Yo la he marcado en azul para distinguirla fácilmente de las demás filas, que contienen los datos propiamente dichos, esto es, los valores de dichos campos, consignando los datos de cada película en cada línea.

Si abrimos esta tabla con *excell* (importar datos csv con el separador **,**), vemos cada dato en una celda.

[Barra de Fórmulas]														
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
actor_3_name	director_name	num_critics	duration	director	actor_1_name	actor_2_name	actor_3_name	movie_title	num_votes	gross	genres	actor_1_facebook_likes	actor_2_facebook_likes	actor_3_facebook_likes
Wes Studi	James Cameron	723	178	0	855 Joel David Moore	1000	76050847	Action Adventure Fantasy Sci-Fi	CCH Pounder	Avatar	886204,4834	Wes Studi	0	Wes Studi
Gore Verbinski	302	169	563	1000	Orlando Bloom	40000	30940152	Action Adventure Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220,48350	Jack Davenport	0	goddes
Sam Mendes	602	148	0	161	Rory Kinnear	11000	20007415	Action Adventure Thriller	Christoph Waltz	Spectre	275868,11700	Stephanie Sigman	1	bomb
Christopher Nolan	813	164	22000	23000	Christian Bale	27000	448130642	Action Thriller	Tom Hardy	The Dark Knight Rises	1144337,106759	Joseph Gordon-L	0	deceptor
Doug Walker		462	132	131	Rob Walker	131		Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens	8,143,0			
Andrew Stanton	462	132	475	530	Samantha Morton	640	73058679	Action Adventure Sci-Fi	Daryl Sabara	John Carter	212204,1873	Polly Walker	1	alien am
Sam Raimi	392	156	0	4000	James Franco	24000	336530303	Action Adventure Romance	J.K. Simmons	Spider-Man 3	383056,46055	Kirsten Dunst	0	sandman
Nathan Greno	324	100	15	284	Donna Murphy	789	20007820	Adventure Animation Comedy Family Fantasy	Brad Garrett	Tangled	294810,2036	M.C. Gainey	1	17th cent
Joss Whedon	635	141	0	15000	Robert Downey Jr.	26000	45899159	Action Adventure Sci-Fi	Chris Hemsworth	Avengers: Age of Ultron	462669,92000	Scarlett Johansson	4	artificial
Dave Yates	375	153	282	10000	Daniel Radcliffe	25000	301596980	Adventure Family Fantasy Mystery	Alan Rickman	Harry Potter and the Half-Blood Prince	321795,5873	Rupert Grint	3	blood bo
Zack Snyder	672	183	0	2000	Ennio Colan	15000	330249602	Action Adventure Sci-Fi	Henry Cavill	Batman v Superman: Dawn of Justice	571839,24450	Alan D. Purwin	0	based on
Bryan Singer	634	169	0	903	Marlon Brando	10000	20009408	Action Adventure Sci-Fi	Kevin Spacey	Superman Returns	240396,29991	Frank Langella	0	crystal le
Marc Forster	403	106	395	393	Mathieu Amalric	451	16836427	Action Adventure	Giancarlo Giannini	Quantum of Solace	333074,2023	Rory Kinnear	1	action he
Gore Verbinski	313	151	563	1000	Orlando Bloom	40000	42302628	Action Adventure Fantasy	Johnny Depp	Pirates of the Caribbean: Dead Man's Chest	522040,44846	Jack Davenport	2	box office
Gore Verbinski	459	150	363	1000	Ruth Wilson	40000	92089910	Action Adventure Western	Johnny Depp	The Lone Ranger	161792,45737	Tom Wilkinson	3	blackbea
Zack Snyder	733	143	0	748	Christopher Meloni	15000	291021563	Action Adventure Fantasy Sci-Fi	Henry Cavill	Man of Steel	548573,2045	Harry Lennix	0	brother b
Andrew Adamson	258	150	80	201	Pierfrancesco Favino	22000	141614023	Action Adventure Fantasy Sci-Fi	Peter Dinklage	The Chronicles of Narnia: Prince Caspian	149922,22697	Damian Alcázar	4	based on
Joss Whedon	703	173	0	15000	Robert Downey Jr.	26000	45899159	Action Adventure Sci-Fi	Chris Hemsworth	The Avengers	462669,92000	Scarlett Johansson	3	alien inv
Rob Marshall	449	136	0	1000	Sam Claflin	40000	241683875	Action Adventure Fantasy	Johnny Depp	Pirates of the Caribbean: On Stranger Tides	370704,54883	Stephen Graham	3	blackbea
Barry Sonnenfeld	451	106	188	718	Michael Schulzberg	10000	179020834	Action Adventure Comedy Family Fantasy	John Smith	Men in Black 3	268154,12572	Nicole Scherzinger	1	alien c
Peter Jackson	422	164	0	773	Adam Brown	5000	255108370	Adventure Fantasy	Aidan Turner	The Hobbit: The Battle of the Five Armies	354228,9152	James Nesbitt	0	army elf
Marc Webb	599	153	464	963	Andrew Garfield	15000	262039063	Action Adventure Fantasy	Emma Stone	The Amazing Spider-Man	451903,26489	Chris Zylka	0	liard ou
Ridley Scott	343	136	0	728	William Hurt	891	192137070	Action Adventure Drama History	Mark Addy	Robin Hood	211765,2344	Scott Grimes	0	alice in
Peter Jackson	509	186	0	773	Adam Brown	5000	25835334	Adventure Fantasy	Aidan Turner	The Hobbit: The Desolation of Smaug	483340,9152	James Nesbitt	6	dwarf el
Chris Weitz	251	113	129	1000	Eva Green	16000	7008319	Adventure Fantasy Drama	Christopher Lee	The Golden Compass	149019,21406	Kirstin Scott Thomas	2	children
Peter Jackson	446	106	0	84	Thomas Kretschmann	6000	210821261	Action Adventure Drama Romance	Benno Für	1122	50	Evans	0	artist li
James Cameron	515	194	0	794	Kate Winslet	29000	658672302	Drama Romance	Leonardo DiCaprio	Titanic	793509,42323	Gloria Stuart	0	based on
Anthony Russo	516	147	94	11000	Scarlett Johansson	21000	471872920	Action Adventure Sci-Fi	Robert Downey Jr.	Captain America: Civil War	727670,64798	Chris Evans	0	based on
Peter Berg	377	131	532	627	Alexander Skarsgård	14000	65173100	Action Adventure Sci-Fi Thriller	Liam Neeson	BattleShip	202262,26679	Tadanobu Asano	0	box office
Colin Trevorrow	444	124	365	1000	Judy Greir	3000	62117271	Action Adventure Sci-Fi Thriller	Brady Corbet	Justice League	41851,2458	Omari N	0	based on
Sam Mendes	705	143	0	380	Helen McCrory	883	304360277	Action Adventure Thriller	Albert Finney	Skyfall	522030,2039	Rory Kinnear	0	brawl ch
Sam Raimi	300	135	0	4000	James Franco	24000	337377893	Action Adventure Fantasy Romance	J.K. Simmons	Spider-Man 2	411164,43386	Kirsten Dunst	1	death to
Shane Black	608	195	1000	3000	Jon Favreau	21000	40892272	Action Adventure Sci-Fi	Robert Downey Jr.	Iron Man 3	567489,34620	Dan Cheadle	3	armor les
Tim Burton	451	118	13000	11000	Alan Rickman	40000	33418326	Adventure Family Fantasy	Johnny Depp	Alice in Wonderland	380219,79057	Anne Hathaway	2	blackbea
Brett Ratner	334	104	420	560	Kelsey Grammer	20000	243480014	Action Adventure Fantasy Sci-Fi Thriller	Hugh Jackman	X-Men: The Last Stand	383427,27174	Daniel Cudmore	0	batte ms
Dan Scanlon	376	104	37	760	Tyler Labine	12000	26488329	Action Adventure Animation Comedy Family Fantasy	Steve Buscemi	Monsters University	322387,3218	Ramon Rodriguez	0	cheatin
Michael Bay	366	150	0	464	Kevin Durand	894	402076602	Action Adventure Sci-Fi	Glenn Morshower	Transformers: Revenge of the Fallen	522307,563	78	235	5000
Michael Bay	378	163	0	808	Sophia Miles	974	245428137	Action Adventure Sci-Fi	Bryanne U	Transformers: Age of Extinction	244240,3988	Kelsey Grammer	0	autobot
Sam Raimi	525	130	0	11000	Mia Kunis	44000	23403076	Adventure Family Fantasy	Tim Holmes	On the Beach and Powerful	175409,73441	James Franco	4	circus m
Marc Webb	495	142	464	825	Andrew Garfield	15000	20263933	Action Adventure Fantasy Sci-Fi	Emma Stone	The Amazing Spider-Man 2	321427,26631	B.J. Novak	0	costumed
Joseph Kosinski	469	125	364	1000	Olivia Wilde	12000	172051787	Action Adventure Sci-Fi	Jeff Bridges	TRON: Legacy	264183,28500	James Frain	0	arcade b

Mostramos la hoja excell en dos imágenes por ser muy ancha.

actor_3_name	director_name	num_critics	duration	keywords	movie_title	num_votes	gross	genres	actor_1_facebook_likes	actor_2_facebook_likes	actor_3_facebook_likes
Wes Studi	James Cameron	723	178	0	Avatar	855	76050847	Action Adventure Fantasy Sci-Fi	CCH Pounder	Wes Studi	0
Gore Verbinski	302	169	563	1000	Pirates of the Caribbean: At World's End	40000	30940152	Action Adventure Fantasy	Johnny Depp	Orlando Bloom	1000
Stephanie Sigman	1	bomb	espionage	sequel	spy	terrorist					
Joseph Gordon-Lewitt	0	deception	imprisonment	lawlessness	police officer	terrorist	plot				
0											
Polly Walker	1	alien	american	civil war	male	nipple	man	princess			
Kirsten Dunst	0	sandman	spider	man	symbolic	venom	villain				
M.C. Gainey	1	17th century	based on	fairly tale	disney	flower	tower				
Scarlett Johansson	4	artificial	intelligence	based on	comic book	captain america	marvel	cinematic	universe	superhero	
Rupert Grint	3	blood	bo	low	potion	professor					
Alan D. Purwin	0	based on	comic book	batman	sequel	to a reboot	superhero	superman			
Frank Langella	0	crystal le	lux	luthor	lois lane	return	to earth				
Rory Kinnear	1	action hero	attempted rape	bond	girl	fictional	character	of james bond	series	revenge	
Jack Davenport	2	box office	hit	planet	equid	hawai	island	king	of the ice	monster	
Tom Wilkinson	1	horse	outlaw	teexas	teexas	ranger	train				
Harry Lennix	0	based on	comic book	british	actor	playing	american	character	final	battle	of origin
Damian Alcázar	4	brother	brother	relationship	brother	sister	relationship	good	versus	evil	king
Stephen Graham	4	blackbeard	captain	pirate	revenge	soldier					
Nicole Scherzinger	1	alien	criminal	m.l.b.	maximum	security	prison	prison			
James Nesbitt	0	army	elf	hobbit	middle	earth	orc				
Chris Zylka	0	liard	outcast	spider	spider	man	teenager				
Scott Grimes	0	113th	archer	king	of england	robin	hood				
James Nesbitt	6	dwarf	elf	lake town	mountain	sword	and	sorcery			
Kirstin Scott Thomas	2	children	epic	friend	girl	quest					
Evan Parke	0	animal	name	in the	lake	abducts	a woman	gorilla	island	king	of kong
Gloria Stuart	0	artist	low	ship	itfanc	leuc					
Chris Evans	0	based on	comic book	knife	marvel	cinematic	universe	returning	character	killed	off
Tadanobu Asano	0	box office	flop	hawaii	hawai	oahu	hawaii	ship			
Omari N	0	dinosaur	disaster	film	experiment	gone	wrong	jurassic	park	velociraptor	
Rory Kinnear	2	brawl	childhood	home	computer	cracker	intelligence	agency	terrorist	cell	
Kirsten Dunst	1	death	doctor	scientist	super	villain	tentacle				
Dan Cheadle	3	armor	outpost	human	bomb	missile	attack	terrorist			
Anne Hathaway	0	alice	in wonderland	impossible	reality	for dream	queen	shrinking	shrinking	potion	
Daniel Cudmore	0	batman	mutant	outage	to a well	leaving	through	a well			
Sean Hayes	0	cheating	fraternity	monster	singing	in a car	university				
Ramon Rodriguez	0	autobot	deception	machine	sun	symbol					
Kelsey Grammer	2	blackboard	bumblebee	the character	is	truck	and	trailer	teexas	truck	
James Franco	4	circus	magic	magician	in the	well					
B.J. Novak	0	costumed	hero	death	of girlfriend	masked	villain	returning	character	killed	off
James Frain	0	arcade	bridge	disappearance	escape	warrior					

Librerías y constantes globales

Pongamos todas las librerías necesarias al principio, tal como propone el estilo `pep-8`. Ej.: [PEP 8 -- Style Guide for Python Code](#).

De paso, en éste y otros lugares de Internet podrás encontrar detalles sobre estilo y presentación de código con un estilo estándar y cuidado. Examina esta página y trata de seguir esas indicaciones.

```
In [68]: # Librerías:

# Esta celda debe ser completada por el estudiante

import csv
import gc
from collections import defaultdict
import matplotlib.pyplot as plt
import requests
from bs4 import BeautifulSoup
import pandas as pd
import seaborn as sns
```

```
In [69]: # Constantes globales:

# Esta celda debe ser completada por el estudiante

MOVIES_DATA = './data_in/movie_data.csv'
CAMPOS_INTERES = {
    'movie_title': 11, 'title_year': 23, 'director_name': 1,
    'actor_1_name': 10, 'language': 19, 'country': 20, 'color': 0,
    'budget': 22, 'imdb_score': 25, 'movie_imdb_link': 17
}
FEW_FIELDS = "algunos_campos.txt"
CAMPOS_REDUCIDOS = ['movie_title', 'color', 'director_name', 'language',
                    'country', 'actor_1_name', 'movie_imdb_link']
OPTIONAL_DATA = './data_in/all_seasons.csv'
```

Parte A. Ejercicios básicos sin usar pandas [2 puntos]

Esta parte inicial debe realizarse sin usar la librería `pandas`. Para practicar con esta librería, se plantean otros apartados más abajo.

A.1. Exploración inicial básica del archivo de datos

Deseamos cargar el archivo de datos, que tiene un formato `csv`. En este apartado, te recomiendo fuertemente usar la librería `cvs`, que deberás importar en la primera celda

del script, más arriba, no aquí. (En los siguientes apartados, ya no mencionaré qué librerías usar ni recordaré dónde se han de importar.)

Observa también que el test de funcionamiento te da el nombre de la función que deber definir y algún otro identificador como es, por ejemplo, la constante `MOVIES_DATA`, que debes definir también más arriba, en la segunda celda de este script.

Finalmente, observando el test, verás que se carga por separado la cabecera y las filas de datos.

```
In [70]: # Esta celda debe ser completada por el estudiante

def load_full_data(archivo_csv: str) -> tuple[list[str], list[list[str]]]:
    """
    Funcion que lee el archivo CSV, separa la cabecera de las filas de datos
    y devuelve ambas como una tupla.

    Parameters
    -----
    archivo_csv : Ruta al archivo CSV desde el que se cargaran los datos.

    Returns
    -----
    (cabecera, datos) : Una tupla con dos elementos:
    - cabecera (list) : Lista de nombres de las columnas.
    - datos (list) : Lista de listas, cada sublista es una fila de datos.
    -----
    """

    with open(archivo_csv, encoding='utf-8') as archivo:
        lector = csv.reader(archivo)
        cabecera = next(lector) # Extraemos la primera fila como cabecera
        datos = [fila for fila in lector] # Extraemos el resto de filas

    return cabecera, datos
```

```
In [71]: # Test de funcionamiento

full_header, full_list_data = load_full_data(MOVIES_DATA)

print(full_header)
print()
print(full_list_data[0:5])
```

```
['color', 'director_name', 'num_critic_for_reviews', 'duration', 'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name', 'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name', 'movie_title', 'num_voted_users', 'cast_total_facebook_likes', 'actor_3_name', 'facenumber_in_poster', 'plot_keywords', 'movie_imdb_link', 'num_user_for_reviews', 'language', 'country', 'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes', 'imdb_score', 'aspect_ratio', 'movie_facebook_likes']
```

```
[[ 'Color', 'James Cameron', '723', '178', '0', '855', 'Joel David Moore', '1000', '760505847', 'Action|Adventure|Fantasy|Sci-Fi', 'CCH Pounder', 'Avatar\xa0', '886204', '4834', 'Wes Studi', '0', 'avatar|future|marine|native|paraplegic', 'http://www.imdb.com/title/tt0499549/?ref=fn_tt_tt_1', '3054', 'English', 'USA', 'PG-13', '237000000', '2009', '936', '7.9', '1.78', '33000'], [ 'Color', 'Gore Verbinski', '302', '169', '563', '1000', 'Orlando Bloom', '40000', '309404152', 'Action|Adventure|Fantasy', 'Johnny Depp', 'Pirates of the Caribbean: At World's End\xa0', '471220', '48350', 'Jack Davenport', '0', 'goddess|marriage ceremony|marriage proposal|pirate|singapore', 'http://www.imdb.com/title/tt0449088/?ref=fn_tt_tt_1', '1238', 'English', 'USA', 'PG-13', '300000000', '2007', '5000', '7.1', '2.35', '0'], [ 'Color', 'Sam Mendes', '602', '148', '0', '161', 'Rory Kinnear', '11000', '200074175', 'Action|Adventure|Thriller', 'Christoph Waltz', 'Spectre\xa0', '275868', '11700', 'Stephanie Sigman', '1', 'bomb|espionage|sequel|spy|terrorist', 'http://www.imdb.com/title/tt2379713/?ref=fn_tt_tt_1', '994', 'English', 'UK', 'PG-13', '245000000', '2015', '393', '6.8', '2.35', '85000'], [ 'Color', 'Christopher Nolan', '813', '164', '22000', '23000', 'Christian Bale', '27000', '448130642', 'Action|Thriller', 'Tom Hardy', 'The Dark Knight Rises\xa0', '1144337', '106759', 'Joseph Gordon-Levitt', '0', 'deception|imprisonment|lawlessness|police officer|terrorist plot', 'http://www.imdb.com/title/tt1345836/?ref=fn_tt_tt_1', '2701', 'English', 'USA', 'PG-13', '250000000', '2012', '23000', '8.5', '2.35', '164000'], [ '', 'Doug Walker', '', '', '131', '', 'Rob Walker', '131', '', 'Documentary', 'Doug Walker', 'Star Wars: Episode VII - The Force Awakens\xa0', '8', '143', '', '0', '', 'http://www.imdb.com/title/tt5289954/?ref=fn_tt_tt_1', '', '', '', '', '', '12', '7.1', '', '0']]
```

Ahora, querríamos ver las posiciones de los identificadores de los campos, dados en la línea cabecera.

```
In [72]: # Esta celda debe ser completada por el estudiante

# Obtenemos las posiciones empleando enumerate,
# para obtener pares con el índice y el encabezado
list_of_enumerated_headers = [(ind, he) for ind, he in enumerate(full_header)]
```

```
In [73]: # Test de funcionamiento

print(list_of_enumerated_headers)
```

```
[(0, 'color'), (1, 'director_name'), (2, 'num_critic_for_reviews'), (3, 'duration'), (4, 'director_facebook_likes'), (5, 'actor_3_facebook_likes'), (6, 'actor_2_name'), (7, 'actor_1_facebook_likes'), (8, 'gross'), (9, 'genres'), (10, 'actor_1_name'), (11, 'movie_title'), (12, 'num_voted_users'), (13, 'cast_total_facebook_likes'), (14, 'actor_3_name'), (15, 'facenumber_in_poster'), (16, 'plot_keywords'), (17, 'movie_imdb_link'), (18, 'num_user_for_reviews'), (19, 'language'), (20, 'country'), (21, 'content_rating'), (22, 'budget'), (23, 'title_year'), (24, 'actor_2_facebook_likes'), (25, 'imdb_score'), (26, 'aspect_ratio'), (27, 'movie_facebook_likes')]
```

A.2. Campos principales de una película

Los campos (columnas) del archivo son demasiados. No nos interesan todos ellos. Dada una lista con todos los campos, se pide extraer otra lista sólo con los campos con los que vamos a trabajar en los siguientes apartados: `movie_title`, `title_year`, `director_name`, `actor_1_name`, `language`, `country`, `color`, `budget`, `imdb_score` y `movie_imdb_link`.

```
In [74]: # Esta celda debe ser completada por el estudiante

def main_data_from_item(lista_datos: list[str]) -> list[str]:
    """
    Función que recibe un lista de datos del archivo, y extrae de la misma los\
    campos de interés indicados en el diccionario CAMPOS_INTERES.

    Parameters
    -----
        lista_datos : Lista de datos de los que extraer los campos de interés.

    Returns
    -----
        lista : Lista con los campos de interes extraidos.
    """

    # Como hemos definido CAMPOS_INTERES como un diccionario, simplemente
    # accedemos a los valores del mismo y devolvemos los campos
    return [lista_datos[ind] for ind in CAMPOS_INTERES.values()]
```

```
In [75]: # Test de funcionamiento

print(main_data_from_item(full_header))

print()

datos_avatar_2009 = main_data_from_item(full_list_data[0])
print(datos_avatar_2009)

print()

datos_star_wars_7 = main_data_from_item(full_list_data[4])
print(datos_star_wars_7)

['movie_title', 'title_year', 'director_name', 'actor_1_name', 'language', 'count
ry', 'color', 'budget', 'imdb_score', 'movie_imdb_link']

['Avatar\xa0', '2009', 'James Cameron', 'CCH Pounder', 'English', 'USA', 'Color',
'237000000', '7.9', 'http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1']

['Star Wars: Episode VII - The Force Awakens\xa0', '', 'Doug Walker',
'Doug Walker', '', '', '', '7.1', 'http://www.imdb.com/title/tt5289954/?ref_=
fn_tt_tt_1']
```

A.3. Algunos ajustes en los campos

Observa la anomalía en el string del título de la película. También, queremos tratar algunos campos como numéricos, ya sea enteros (el año y el presupuesto) o reales (la valoración). (Algunos campos numéricos enteros están vacíos en el archivo de datos;

para ellos, el valor imputado será `-1`.) Además, en las urls de las películas no necesitaremos el fragmento final, iniciado con `?ref_`.

```
In [76]: # Esta celda debe ser completada por el estudiante

def convierteNumero(valor: str, esInt: bool) -> int:
    """
    Función que recibe un string que representa un número y, en función de un\
    parámetro que determina si es entero o float, lo convierte al tipo deseado.\
    En caso de que el valor recibido no sea un número, devolverá -1.

    Parameters
    -----
        valor : El número a convertir, que vendrá en formato string.
        esInt : Bool, es True si el número es entero, y False si es decimal.

    Returns
    -----
        valor : El número, convertido a entero/float, según sea el caso; o -1.
    """

    try:
        return int(valor) if esInt else float(valor)
    except ValueError:
        return -1

def cleanTitle(titulo: str) -> str:
    """
    Función que recibe un string, representando el título de una película, y \
    le elimina los caracteres sobrantes al final.

    Parameters
    -----
        titulo : El string que queremos limpiar.

    Returns
    -----
        valor : El string, con los caracteres finales '\\xa0' eliminados.
    """

    return titulo.strip().replace('\\xa0', '')

def correctURL(url: str) -> str:
    """
    Función que recibe un string, representando la url en IMDB de una película\
    y la devuelve sin el fragmento final iniciado por '?ref_'.

    Parameters
    -----
        titulo : El string con la URL que queremos limpiar.

    Returns
    -----
        valor : El string, con los caracteres finales eliminados.
    """

    return url.strip().split('?ref')[0]

def datatypes_arranged(lista_datos : list[str]) -> list[str]:
```



```

"""
Función que recibe una lista con todos los datos de la película y los \
corrige, modificando los tipos de datos de los valores numéricos y \
limpiando el título y la url de la misma.

Parameters
-----
    lista_datos : Una lista con todos los campos de datos de la película.

Returns
-----
    lista_datos : La lista, pero con los valores pertinentes corregidos.
"""

lista_datos[0] = cleanTitle(lista_datos[0]) # Limpiamos el título
lista_datos[1] = convierteNumero(lista_datos[1], True) # Año: int
lista_datos[7] = convierteNumero(lista_datos[7], True) # Presupuesto: int
lista_datos[8] = convierteNumero(lista_datos[8], False) # Puntuación: float
lista_datos[9] = correctURL(lista_datos[9]) # Limpiamos la url

return lista_datos

```

In [77]: *# Test de funcionamiento*

```

print(datatypes_arranged(datos_avatar_2009))

print(datatypes_arranged(datos_star_wars_7))

```

```

['Avatar', 2009, 'James Cameron', 'CCH Pounder', 'English', 'USA', 'Color', 23700
0000, 7.9, 'http://www.imdb.com/title/tt0499549/']
['Star Wars: Episode VII - The Force Awakens', -1, 'Doug Walker', 'Doug Walker',
'', '', '', -1, 7.1, 'http://www.imdb.com/title/tt5289954/']

```

A.4. Recuperación de alguna información básica

Diseña funciones para recuperar la siguiente información:

- El conjunto de valores posibles del campo `Color`.
- Los títulos de película de nuestro archivo (limpios de caracteres extraños), junto con la cantidad de calificadores (un entero), que contengan la subcadena "Victor".

In [78]: *# Esta celda debe ser completada por el estudiante*

```

def getColores(lista_peliculas: list[list[str]]) -> set[str]:
    """
    Extrae los colores únicos de una lista de películas.

    Parameters
    -----
        lista_peliculas : Una lista de listas de strings, donde cada lista \
            son los datos de una película, y cada sublista uno de los campos.

    Returns
    -----
        valor : Un conjunto con los valores únicos del campo color.
    """

```



```

colores = set()
for movie in lista_peliculas:
    colores.add(movie[0])
return colores

def getPelisVictor(lista_peliculas: list[list[str]]) -> list[tuple[str, int]]:
    """
    Extrae los títulos de las películas, limpios de caracteres extraños, junto\
    con los calificadores, de los títulos que contienen la subcadena "Victor".

    Parameters
    -----
    lista_peliculas : Una lista de listas de strings, donde cada lista \
    son los datos de una película, y cada sublista uno de los campos.

    Returns
    -----
    victors : Una lista de tuplas, donde el primer elemento es el título \
    de la película, y el segundo, el número de usuarios que la calificaron.
    """

    victors = list()
    for movie in lista_peliculas:
        if "Victor" in cleanTitle(movie[11]):
            victors.append((cleanTitle(movie[11]), int(movie[2])))

    return victors

colores = getColores(full_list_data)
pelis_victor = getPelisVictor(full_list_data)

```

In [79]: *# Test de funcionamiento*

```

print(colores)
print(pelis_victor)

```

```

{'', 'Color', ' Black and White'}
[('Victor Frankenstein', 159), ('Victor Frankenstein', 159), ('The Young Victori
a', 188), ('Victor Frankenstein', 159), ('Raising Victor Vargas', 59)]

```

B Datos en un diccionario [2 puntos]

B.1. Carga únicamente de datos que vamos a usar

En lugar de cargar *todos* los datos del archivo, luego generar otro archivo con los datos que nos interesan y luego ajustarlos, es mejor diseñar una función que recorre el archivo de datos y, de cada línea del archivo, *únicamente* retiene la parte que nos interesa, cargando la cabecera por un lado y, por otro, la lista de datos de interés, ajustados como ya hemos dicho.

Cargaremos esta información en un diccionario, cuyas claves serán pares (título, año) y cuyos valores se recogen en una lista con el resto de los campos.

Además, los datos de cada película tras la clave (título, año) están repetidas en el archivo de datos, con todos los datos idénticos; y lo que es peor: alguna que está

repetida, pero con algún dato distinto. En ambos casos, para simplificar, adoptamos el criterio de registrar únicamente la primera aparición de los datos de una misma película y año.

```
In [80]: # Esta celda debe ser completada por el estudiante

def load_main_data(
    ruta_archivo: str
) -> tuple[list[str], dict[tuple[str, int], list[str]]]:
    """
    Lee el archivo CSV indicado en ruta_archivo y guarda los datos en un \
    diccionario, cuya clave es una tupla (título, año). En caso de una clave \
    duplicada, se conserva el primer registro. Devuelve este diccionario, así \
    como una lista con los encabezados de los campos.

    Parameters
    -----
        ruta_archivo : Un string que representa la ruta al archivo CSV.

    Returns
    -----
        list[str] : Una lista con los nombres de los campos del archivo.
        dict : Un diccionario cuya clave es una tupla (título, año) y sus \
        valores una lista de strings que representan los campos.
    """

    # Los campos de interés podemos extraerlos directamente de la variable
    campos_interes = list(CAMPOS_INTERES.keys())
    peliculas = {}

    with open(ruta_archivo, encoding='utf-8') as archivo:
        lector = csv.DictReader(archivo)
        for row in lector: # Leemos una línea del archivo
            # La clave sera el título, limpio; y el año, convertido a int
            key = (cleanTitle(row['movie_title']),
                  convierteNumero(row['title_year'], True))
            if key not in peliculas: # Si la película no está, la añadimos
                datos = [row[campo] for campo in campos_interes
                          if campo not in ['movie_title', 'title_year']]
                # Convertimos los tipos de los campos correspondientes
                datos[5] = convierteNumero(datos[5], True) # Presupuesto
                datos[6] = convierteNumero(datos[6], False) # Puntuación
                datos[7] = correctURL(datos[7]) # Limpiamos la url
                peliculas[key] = datos # Asignamos los valores a la clave

    return campos_interes, peliculas
```

```
In [81]: # Test de funcionamiento

main_header, main_dict_data = load_main_data(MOVIES_DATA)

print(main_header)
print()
for title_year, pieces in list(main_dict_data.items())[:5]:
    print(title_year, " -> ", pieces)
```

```
['movie_title', 'title_year', 'director_name', 'actor_1_name', 'language', 'country', 'color', 'budget', 'imdb_score', 'movie_imdb_link']
```

```
('Avatar', 2009) -> ['James Cameron', 'CCH Pounder', 'English', 'USA', 'Color', 237000000, 7.9, 'http://www.imdb.com/title/tt0499549/']
('Pirates of the Caribbean: At World's End', 2007) -> ['Gore Verbinski', 'Johnny Depp', 'English', 'USA', 'Color', 300000000, 7.1, 'http://www.imdb.com/title/tt0449088/']
('Spectre', 2015) -> ['Sam Mendes', 'Christoph Waltz', 'English', 'UK', 'Color', 245000000, 6.8, 'http://www.imdb.com/title/tt2379713/']
('The Dark Knight Rises', 2012) -> ['Christopher Nolan', 'Tom Hardy', 'English', 'USA', 'Color', 250000000, 8.5, 'http://www.imdb.com/title/tt1345836/']
('Star Wars: Episode VII - The Force Awakens', -1) -> ['Doug Walker', 'Doug Walker', '', '', '', -1, 7.1, 'http://www.imdb.com/title/tt5289954/']
```

Y ahora podemos suprimir la variable `full_list_data`

```
In [82]: # Esta celda debe ser completada por el estudiante

# Eliminamos la variable y llamamos al garbage collector
del full_list_data
gc.collect()
```

Out[82]: 77045

```
In [83]: # Test de funcionamiento

try:
    print(full_list_data)
except:
    print('La variable full_list_data está suprimida correctamente')
```

La variable `full_list_data` está suprimida correctamente

B.2. Recuperación de alguna información

Diseña funciones para averiguar la siguiente información:

- ¿Qué títulos de películas han sido dirigidas por "James Cameron" (o por el director que se desee)?

```
In [84]: # Esta celda debe ser completada por el estudiante

def movies_anno_for_director(
    diccionario_peliculas: dict[tuple[str, int], list[str]],
    director: str
) -> list[tuple[str, int]]:
    """
    Recibe un diccionario, cuya clave es una tupla (título, año) y una cadena\
    de texto que representa un director, y devuelve las películas (la clave\
    del diccionario) que ha dirigido ese director.

    Parameters
    -----
    diccionario_peliculas : Un diccionario con todas las películas.
    director : Nombre del director para el que se obtendrán las películas.

    Returns
```

```

-----
list[tuple[str, int]] : Una lista con las tuplas (título, año) de \
las películas dirigidas por el director.
"""

peliculas = []

for pelicula, datos in diccionario_peliculas.items():
    if datos[0] == director:
        peliculas.append(pelicula)

return peliculas

```

In [85]: *# Test de funcionamiento*

```
movies_anno_for_director(main_dict_data, "James Cameron")
```

Out[85]: [('Avatar', 2009),
('Titanic', 1997),
('Terminator 2: Judgment Day', 1991),
('True Lies', 1994),
('The Abyss', 1989),
('Aliens', 1986),
('The Terminator', 1984)]

- ¿Qué directores han dirigido el número máximo de películas?

In [86]: *# Esta celda debe ser completada por el estudiante*

```

def directors_max_movies(
    diccionario_peliculas: dict[tuple[str, int], list[str]]
) -> tuple[list[str], int]:
    """
    Recibe un diccionario, cuya clave es una tupla (título, año) y halla el \
    director (o directores) que han dirigido el mayor número de películas, así\
    como dicho máximo.

    Parameters
    -----
    diccionario_peliculas : Un diccionario con todas las películas.

    Returns
    -----
    tuple[list[str], int] : Una tupla con una lista con el nombre del \
    director o directores, y el número máximo de películas dirigidas.
    """

    directores={}

    # En primer lugar, hallamos para todos los directores, el número de
    # películas que han dirigido
    for pelicula in diccionario_peliculas.values():
        director = pelicula[0]
        # Si el director está en la lista, añadimos una película;
        # si no lo está, lo añadimos a la lista
        if director != '' and director in directores:
            directores[director] += 1
        else:
            directores[director] = 1

```

```
# Obtenemos el máximo número de películas dirigidas por un mismo director
maximo = max(directores.values())
# Recorremos la lista de directores para encontrar
# aquel(los) que coincidan con el máximo
director = [dir for dir, count in directores.items() if count==maximo]

return (director, maximo)
```

In [87]: *# Test de funcionamiento*

```
print(directors_max_movies(main_dict_data))
```

(['Steven Spielberg'], 26)

- Para cada año de un intervalo dado de años, ¿cuántas películas se han realizado? Esta información debe recuperarse en un diccionario convencional (no por defecto), y luego debe mostrarse en una lista con los años en orden ascendente..

In [88]: *# Esta celda debe ser completada por el estudiante*

```
def years_num_movies(
    diccionario_peliculas: dict[tuple[str, int], list[str]],
    inicio: int,
    fin: int
) -> dict[int, int]:
    """
    Recibe un diccionario, cuya clave es una tupla (título, año), un año de\
    inicio y un año de fin y devuelve un diccionario cuyas claves son los años\
    y sus valores el número de películas realizadas en ese año.

    Parameters
    -----
        diccionario_peliculas : Un diccionario con todas las películas.
        inicio : El año de inicio del intervalo.
        fin : El año de fin del intervalo.

    Returns
    -----
        dict[int, int] : Un diccionario cuyas claves son los años, y sus \
        valores el número de películas realizadas.
    """

    # Inicializamos el diccionario con los años en el rango (inicio, fin) y
    # el conteo de películas a 0. Al hacerlo así, garantizamos además que
    # los años salgan ordenados en orden ascendente
    anhos = {anho:0 for anho in range(inicio, fin+1)}

    # Recorremos el diccionario de películas
    for pelicula in diccionario_peliculas.keys():
        anho_pelicula = pelicula[1] # El año es el segundo parámetro
        # Si el año de la película está entre el año de inicio y el de fin
        if inicio <= anho_pelicula <= fin:
            # Incrementamos el contador de ese año en una unidad
            anhos[anho_pelicula] += 1

    return anhos
```

In [89]: *# Test de funcionamiento*

```
num_movies = years_num_movies(main_dict_data, 2000, 2015)

print(num_movies)
```

```
{2000: 169, 2001: 183, 2002: 204, 2003: 169, 2004: 207, 2005: 216, 2006: 235, 2007: 198, 2008: 223, 2009: 253, 2010: 225, 2011: 224, 2012: 214, 2013: 231, 2014: 243, 2015: 211}
```

In [90]: *# Esta celda debe ser completada por el estudiante*

```
# Como la función diseñada ya devuelve los resultados por orden,
# simplemente la llamamos.
num_movies_sorted = years_num_movies(main_dict_data, 2000, 2015)
```

In [91]: *# Test de funcionamiento*

```
print(num_movies_sorted)
```

```
{2000: 169, 2001: 183, 2002: 204, 2003: 169, 2004: 207, 2005: 216, 2006: 235, 2007: 198, 2008: 223, 2009: 253, 2010: 225, 2011: 224, 2012: 214, 2013: 231, 2014: 243, 2015: 211}
```

- Diseña también una operación que, partiendo de nuestro diccionario de los datos principales, almacene en un archivo los datos siguientes de cada película, sin la cabecera: el título, el idioma, el año, el país y el presupuesto. El separador será en este caso el carácter `|`. El archivo usado para el almacenamiento es el siguiente:

```
FEW_FIELDS = "algunos_campos.txt"
```

In [92]: *# Esta celda debe ser completada por el estudiante*

```
def store_file(
    diccionario_peliculas: dict[tuple[str, int], list[str]],
    archivo_destino: str
) -> None:
    """
    Recibe un diccionario, cuya clave es una tupla (título, año), y un \
    string, que representa la ruta a un archivo, e imprime en dicho archivo \
    los campos **título**, **idioma**, **año**, **país** y **presupuesto** \
    de la película, separados por |.

    Parameters
    -----
    diccionario_peliculas : Un diccionario con todas las películas.
    archivo_destino : La ruta al archivo donde se escribirán los datos.

    Returns
    -----
    None
    """

    with open(archivo_destino, "w") as archivo: # Abrimos en modo escritura
        for pelicula in diccionario_peliculas.items():
            # El título y el año son la clave, el primer elemento
            titulo = pelicula[0][0]
            anho = str(pelicula[0][1])
```

```

# El resto de datos están en el siguiente elemento
idioma = pelicula[1][2]
pais = pelicula[1][3]
presupuesto = str(pelicula[1][5])
# Generamos la línea que escribiremos
out = (titulo + '|' + anho + '|' + idioma + '|' + pais +
'|' + presupuesto + '\n')

archivo.write(out)

```

In [93]: *# Test de funcionamiento*

```

store_file(main_dict_data, FEW_FIELDS)

! dir algunos*. *

print()

with open(FEW_FIELDS) as f:
    for i in range(5):
        print(f.readline())

```

El volumen de la unidad C no tiene etiqueta.

El número de serie del volumen es: 98FE-7047

Directorio de c:\Users\Hugo\OneDrive\Desktop\Entregas\EntregasMaster\01. Python
 \Gomez_Sabucedo_Hugo

```

14/11/2024  21:14          210.855 algunos_campos.txt
              1 archivos          210.855 bytes
              0 dirs  484.433.223.680 bytes libres

```

Avatar|2009|English|USA|237000000

Pirates of the Caribbean: At World's End|2007|English|USA|300000000

Spectre|2015|English|UK|245000000

The Dark Knight Rises|2012|English|USA|250000000

Star Wars: Episode VII - The Force Awakens|-1|||-1

B.3. Un conteo sencillo con `defaultdict`

Deseamos saber con qué directores y número de veces ha actuado cada actor como actor principal. Se pide realizar este conteo en un `defaultdict` cuyas claves serán nombres de los actores y cuyos valores tendrán la estructura de un `defaultdict`, cuyas claves serán los nombres de los directores y cuyos valores serán los números contabilizados.

In [94]: *# Esta celda debe ser completada por el estudiante*

```

def actor_directors(
    diccionario_peliculas: dict[tuple[str, int], list[str]]
) -> defaultdict:
    """

```


Recibe un diccionario, cuya clave es una tupla (título, año) e imprime, \ para cada actor, con qué directores ha actuado como actor principal y el \ número de veces que lo ha hecho. Para ello, crea un defaultdict cuya \ clave es el nombre del actor y sus valores son otro defaultdict, con las \ claves siendo el nombre del director y los valores el conteo.

Parameters

diccionario_peliculas : Un diccionario con todas las películas.

Returns

actores : Un defaultdict con clave el nombre del actor y valor otro \ defaultdict con clave el nombre del director y valor el número de \ actuaciones.

"""

```
actores = defaultdict(lambda: defaultdict(int))
```

```
for clave, datos in diccionario_peliculas.items():
    # El primer campo (0) es el director, el segundo (1) el actor
    actores[datos[1]][datos[0]] += 1
```

```
return actores
```

In [95]: *# Test de funcionamiento*

```
num_collaborations = actor_directors(main_dict_data)

print(type(num_collaborations))
key_a, value_a = list(num_collaborations.items())[0]
print(type(key_a), type(value_a))
key_b, value_b = list(value_a.items())[0]
print(type(key_b), type(value_b))

print()

print(num_collaborations)
```

```
<class 'collections.defaultdict'>
<class 'str'> <class 'collections.defaultdict'>
<class 'str'> <class 'int'>
```

```
defaultdict(<function actor_directors.<locals>.<lambda> at 0x0000019548739A80>,
{'CCH Pounder': defaultdict(<class 'int'>, {'James Cameron': 1, 'Peter Hyams': 1,
'Fred Dekker': 1, 'Ernest R. Dickerson': 1}), 'Johnny Depp': defaultdict(<class
'int'>, {'Gore Verbinski': 5, 'Rob Marshall': 2, 'Tim Burton': 6, 'James Bobin':
1, 'Michael Mann': 1, 'Florian Henckel von Donnersmarck': 1, 'Wally Pfister': 1,
'David Koepp': 2, 'Scott Cooper': 1, 'Albert Hughes': 1, 'Roman Polanski': 1, 'We
s Craven': 1, 'Mike Newell': 1, 'Rand Ravich': 1, 'Ted Demme': 1, 'Robert Rodrigu
ez': 1, 'Marc Forster': 1, 'Jeremy Leven': 1, 'Laurence Dunmore': 1, 'Terry Gilli
am': 1, 'Lasse Hallström': 1, 'Oliver Stone': 1, 'Rachel Talalay': 1, 'Kevin Smit
h': 2}), 'Christoph Waltz': defaultdict(<class 'int'>, {'Sam Mendes': 1, 'David Y
ates': 1, 'Michel Gondry': 1, 'Tim Burton': 1}), 'Tom Hardy': defaultdict(<class
'int'>, {'Christopher Nolan': 1, 'George Miller': 1, 'McG': 1, 'Stuart Baird': 1,
'Daniel Espinosa': 1, 'Brian Helgeland': 1, 'Gavin O'Connor': 1, 'Guy Ritchie':
1, '': 1, 'Matthew Vaughn': 1, 'Nicolas Winding Refn': 1}), 'Doug Walker': defaul
tdict(<class 'int'>, {'Doug Walker': 1}), 'Daryl Sabara': defaultdict(<class 'in
t'>, {'Andrew Stanton': 1, 'Eli Roth': 1}), 'J.K. Simmons': defaultdict(<class 'i
nt'>, {'Sam Raimi': 4, 'Alan Taylor': 1, 'Alessandro Carloni': 1, 'Tom McGrath':
1, 'Joe Johnston': 1, 'Michael Caton-Jones': 1, 'John Hamburg': 1, 'Gore Verbinsk
i': 1, 'Joan Chen': 1, 'Michael Apted': 1, 'Ethan Coen': 2, 'Jason Reitman': 4,
'Gavin Hood': 1, 'Baltasar Kormákur': 1, 'Karyn Kusama': 1, 'Vicky Jenson': 1, 'M
ichael Polish': 1, 'Jonas Elmer': 1, 'Mike Judge': 1, 'Brian Klugman': 1, 'Damien
Chazelle': 1, 'David Ayer': 1}), 'Brad Garrett': defaultdict(<class 'int'>, {'Nat
han Greno': 1, 'Marc Lawrence': 1}), 'Chris Hemsworth': defaultdict(<class 'in
t'>, {'Joss Whedon': 2, 'Rupert Sanders': 1, 'Alan Taylor': 1, 'Kenneth Branagh':
1, 'J.J. Abrams': 1, 'Cedric Nicolas-Troyan': 1, 'Ron Howard': 2, 'Michael Mann':
1, 'John Francis Daley': 1, 'Drew Goddard': 1, 'David Twohy': 1, 'Stephen Milburn
Anderson': 1}), 'Alan Rickman': defaultdict(<class 'int'>, {'David Yates': 1, 'Ke
vin Reynolds': 1, 'Dean Parisot': 1, 'John McTiernan': 1, 'Neil Jordan': 1, 'Simo
n Wincer': 1, 'Ang Lee': 1, 'Randall Miller': 1}), 'Henry Cavill': defaultdict(<c
lass 'int'>, {'Zack Snyder': 2, 'Guy Ritchie': 1, 'Tarsem Singh': 1, 'Matthew Vau
ghn': 1, 'Kevin Reynolds': 1, 'Mabrouk El Mechri': 1}), 'Kevin Spacey': defaulttdi
ct(<class 'int'>, {'Bryan Singer': 2, 'Wolfgang Petersen': 1, 'F. Gary Gray': 1,
'Alan Parker': 1, 'Iain Softley': 1, 'John Lasseter': 1, 'Joel Schumacher': 1, 'S
ean Anders': 1, 'Mimi Leder': 1, 'Seth Gordon': 1, 'Robert Luketic': 1, 'Curtis H
anson': 1, 'Clint Eastwood': 1, 'Lasse Hallström': 1, 'Grant Heslov': 1, 'Kevin S
pacey': 1, 'Sam Mendes': 1, 'James Foley': 1, 'George Hickenlooper': 1, 'Duncan J
ones': 1, 'J.C. Chandor': 1}), 'Giancarlo Giannini': defaultdict(<class 'int'>,
{'Marc Forster': 1}), 'Peter Dinklage': defaultdict(<class 'int'>, {'Andrew Adams
on': 1, 'Steve Martino': 1, 'Chris Columbus': 1, 'Clay Kaytis': 1, 'Jon Favreau':
1, 'Ben Falcone': 1, 'Frank Oz': 1, 'Sidney Lumet': 1, 'Hue Rhodes': 1, 'Joseph K
osinski': 1, 'Tom McCarthy': 1, 'Eric Schaeffer': 1}), 'Will Smith': defaultdict
(<class 'int'>, {'Barry Sonnenfeld': 4, 'David Ayer': 1, 'Peter Berg': 1, 'Franci
s Lawrence': 1, 'Michael Bay': 2, 'M. Night Shyamalan': 1, 'Michael Mann': 1, 'Al
ex Proyas': 1, 'Tony Scott': 1, 'Roland Emmerich': 1, 'Glenn Ficarra': 1, 'Andy T
ennant': 1, 'Gabriele Muccino': 2, 'Peter Landesman': 1}), 'Aidan Turner': defaul
tdict(<class 'int'>, {'Peter Jackson': 3, 'Harald Zwart': 1}), 'Emma Stone': defa
ultdict(<class 'int'>, {'Marc Webb': 2, 'Tom Dey': 1, 'Cameron Crowe': 1, 'Tate T
aylor': 1, 'Fred Wolf': 1, 'Ruben Fleischer': 1, 'Alejandro G. Iñárritu': 1, 'Gre
g Mottola': 1, 'Peter Cattaneo': 1, 'Will Gluck': 1}), 'Mark Addy': defaultdict(<
class 'int'>, {'Ridley Scott': 1, 'Simon Wells': 1, 'Richard J. Lewis': 1}), 'Chr
istopher Lee': defaultdict(<class 'int'>, {'Chris Weitz': 1, 'Peter Jackson': 2,
'Joe Dante': 1, 'Steven Spielberg': 1, 'James Fargo': 1, 'Alan Metter': 1, 'Guy H
amilton': 1, 'Harald Reinl': 1}), 'Naomi Watts': defaultdict(<class 'int'>, {'Pet
er Jackson': 1, 'Robert Schwentke': 1, 'Hideo Nakata': 1, 'Tom Tykwer': 1, 'Jim S
heridan': 1, 'Doug Liman': 1, 'J.A. Bayona': 1, 'Rachel Talalay': 1, 'David O. Ru
ssell': 1, 'Alejandro G. Iñárritu': 1, 'David Lynch': 1, 'Michael Haneke': 1, 'No
```

```

ah Baumbach': 1, 'Rodrigo García': 1}}, 'Leonardo DiCaprio': defaultdict(<class
'int'>, {'James Cameron': 1, 'Baz Luhrmann': 2, 'Christopher Nolan': 1, 'Alejandr
o G. Iñárritu': 1, 'Martin Scorsese': 5, 'Quentin Tarantino': 1, 'Edward Zwick':
1, 'Ridley Scott': 1, 'Steven Spielberg': 1, 'Danny Boyle': 1, 'Sam Mendes': 1,
'Randall Wallace': 1, 'Clint Eastwood': 1, 'Sam Raimi': 1, 'Jerry Zaks': 1}}, 'Ro
bert Downey Jr.': defaultdict(<class 'int'>, {'Anthony Russo': 1, 'Shane Black':
2, 'Jon Favreau': 2, 'Guy Ritchie': 2, 'Ben Stiller': 1, 'David Fincher': 1, 'Tod
d Phillips': 1, 'Brian Robbins': 1, 'Joe Wright': 1, 'Frank Oz': 1, 'Curtis Hanso
n': 2, 'David Dobkin': 1, 'Mathieu Kassovitz': 1, 'Neil Jordan': 1, 'Jodie Foste
r': 1, 'Richard Linklater': 1, 'Steven Shainberg': 1, 'Jon Poll': 1, 'Richard Lon
craine': 1, 'Keith Gordon': 1, 'George Clooney': 1, 'James Toback': 1}}, 'Liam Ne
eson': defaultdict(<class 'int'>, {'Peter Berg': 1, 'Jonathan Liebesman': 1, 'Lou
is Leterrier': 1, 'Ridley Scott': 1, 'Kathryn Bigelow': 1, 'Jan de Bont': 1, 'Set
h MacFarlane': 2, 'Jaume Collet-Serra': 3, 'Olivier Megaton': 2, 'Peter Yates':
1, 'Peter Lepeniotis': 1, 'Michael Caton-Jones': 1, 'Scott Frank': 1, 'Pierre Mor
el': 1, 'Steven Spielberg': 1, 'Joe Carnahan': 1, 'Roger Donaldson': 1, 'Anthony
Silverston': 1, 'Atom Egoyan': 1, 'John H. Lee': 1, 'Bill Condon': 1, 'Eric Blake
ney': 1, 'Agnieszka Wojtowicz-Vosloo': 1}}, 'Bryce Dallas Howard': defaultdict(<c
lass 'int'>, {'Colin Trevorrow': 1, 'M. Night Shyamalan': 2, 'Lars von Trier': 1,
'David Lowery': 1, 'Marianna Palka': 1}}, 'Albert Finney': defaultdict(<class 'in
t'>, {'Sam Mendes': 1, 'Tony Richardson': 1}}, 'Hugh Jackman': defaultdict(<class
'int'>, {'Brett Ratner': 1, 'Gavin Hood': 1, 'David Bowers': 1, 'Joe Wright': 1,
'Peter Ramsey': 1, 'Bryan Singer': 2, 'James Mangold': 2, 'Shawn Levy': 1, 'Domin
ic Sena': 1, 'Tom Hooper': 1, 'Neill Blomkamp': 1, 'Denis Villeneuve': 1, 'Darren
Aronofsky': 1, 'Dexter Fletcher': 1, 'Tony Goldwyn': 1, 'Elizabeth Banks': 1}},
'Steve Buscemi': defaultdict(<class 'int'>, {'Dan Scanlon': 1, 'Hironobu Sakaguch
i': 1, 'Pete Docter': 1, 'Will Finn': 1, 'Genndy Tartakovsky': 2, 'Gary Winick':
1, 'Dennis Dugan': 3, 'Simon West': 1, 'Gil Kenan': 1, 'Tim Burton': 1, 'Harold B
ecker': 1, 'Steven Brill': 1, 'John Carpenter': 1, 'Betty Thomas': 1, 'Robert Rod
riguez': 1, 'Don Scardino': 1, 'Robert Altman': 1, 'Miguel Arteta': 1, 'Gary Fled
er': 1, 'Chris Rock': 1, 'Olivier Assayas': 1, 'Oren Moverman': 1, 'Jonathan Wack
s': 1, 'Steve Buscemi': 1}}, 'Glenn Morshower': defaultdict(<class 'int'>, {'Mich
ael Bay': 2, 'Jon Amiel': 1}}, 'Bingbing Li': defaultdict(<class 'int'>, {'Michae
l Bay': 1, 'Li Zhang': 1, 'Wayne Wang': 1}}, 'Tim Holmes': defaultdict(<class 'in
t'>, {'Sam Raimi': 1}}, 'Jeff Bridges': defaultdict(<class 'int'>, {'Joseph Kosin
ski': 1, 'Ash Brannon': 1, 'Sergey Bodrov': 1, 'Gary Ross': 1, 'Mark Osborne': 1,
'Michael Cimino': 1, 'Barbra Streisand': 1, 'Ridley Scott': 1, 'Phillip Noyce':
1, 'Terry Gilliam': 1, 'Mark Pellington': 1, 'Jessica Bendinger': 1, 'Albert Broo
ks': 1, 'Michael Meredith': 1, 'Rod Lurie': 1, 'Scott Cooper': 1, 'Larry Charle
s': 1}}, 'Joe Mantegna': defaultdict(<class 'int'>, {'John Lasseter': 1, 'Jon Avn
et': 1, 'Patrick Read Johnson': 1, 'John Schlesinger': 1, 'Tom Holland': 1, 'Stua
rt Gordon': 1, 'Charlie Levi': 1}}, 'Ryan Reynolds': defaultdict(<class 'int'>,
{'Martin Campbell': 1, 'David Soren': 1, 'Robert Schwentke': 1, 'Kirk De Micco':
1, 'David S. Goyer': 1, 'Tim Miller': 1, 'David Dobkin': 1, 'Anne Fletcher': 1,
'Tarsem Singh': 1, 'Joe Carnahan': 1, 'Andrew Fleming': 1, 'Simon Curtis': 1, 'Ad
am Brooks': 1, 'Walt Becker': 1, 'Rodrigo Cortés': 1, 'Rob McKittrick': 1}}, 'Tom
Hanks': defaultdict(<class 'int'>, {'Lee Unkrich': 1, 'Robert Zemeckis': 3, 'Ron
Howard': 4, 'Tom Tykwer': 1, 'John Lasseter': 2, 'Sam Mendes': 1, 'Steven Spielbe
rg': 3, 'Nora Ephron': 1, 'Frank Darabont': 1, 'Paul Greengrass': 1, 'Penny Marsh
all': 2, 'Stephen Daldry': 1, 'John Lee Hancock': 1, 'Tom Hanks': 2}}, 'Christian
Bale': defaultdict(<class 'int'>, {'McG': 1, 'Christopher Nolan': 3, 'Ridley Scot
t': 1, 'Yimou Zhang': 1, 'Rob Bowman': 1, 'John Madden': 1, 'Mike Gabriel': 1, 'J
ohn Singleton': 1, 'James Mangold': 1, 'Terrence Malick': 1, 'Hayao Miyazaki': 1,
'Scott Cooper': 1, 'Todd Haynes': 2, 'Kurt Wimmer': 1, 'Gillian Armstrong': 1, 'D
avid O. Russell': 1, 'Werner Herzog': 1, 'Mary Harron': 1, 'Brad Anderson': 1}},
'Jason Statham': defaultdict(<class 'int'>, {'James Wan': 1, 'Simon West': 3, 'Pa
trick Hughes': 1, 'Sylvester Stallone': 1, 'Gary McKendry': 1, 'Paul W.S. Anderso
n': 1, 'F. Gary Gray': 1, 'Uwe Boll': 1, 'James Wong': 1, 'Taylor Hackford': 1,
'Boaz Yakin': 1, 'Louis Leterrier': 2, 'John Carpenter': 1, 'Philip G. Atwell':
1, 'Gary Fleder': 1, 'Roger Donaldson': 1, 'Mark Neveldine': 2, 'Guy Ritchie': 2,

```

```

'Barry Skolnick': 1, 'Hunter Richards': 1}), 'Peter Capaldi': defaultdict(<class
'int'>, {'Marc Forster': 1}), 'Jennifer Lawrence': defaultdict(<class 'int'>, {'B
ryan Singer': 2, 'Matthew Vaughn': 1, 'Francis Lawrence': 3, 'Gary Ross': 1, 'Dav
id O. Russell': 3, 'Jodie Foster': 1, 'Mark Tonderai': 1, 'Debra Granik': 1, 'Lor
i Petty': 1, 'Drake Doremus': 1}), 'Benedict Cumberbatch': defaultdict(<class 'in
t'>, {'J.J. Abrams': 1, 'Eric Darnell': 1, 'Joe Wright': 1, 'Bill Condon': 1, 'Jo
hn Wells': 1, 'Tomas Alfredson': 1, 'Morten Tyldum': 1}), 'Eddie Marsan': default
dict(<class 'int'>, {'Bryan Singer': 1, 'Mike Leigh': 1}), 'Jake Gyllenhaal': def
aultdict(<class 'int'>, {'Mike Newell': 1, 'Roland Emmerich': 1, 'Sam Mendes': 1,
'Duncan Jones': 1, 'Antoine Fuqua': 1, 'Edward Zwick': 1, 'David O. Russell': 1,
'Brad Silberling': 1, 'David Ayer': 1, 'Ang Lee': 1, 'Blair Hayes': 1, 'Dan Gilro
y': 1, 'Miguel Arteta': 1, 'Richard Kelly': 1, 'James Cox': 1}), 'Charlie Hunna
m': defaultdict(<class 'int'>, {'Guillermo del Toro': 1, 'Alfonso Cuarón': 1, 'St
ephon Gaghan': 1, 'Douglas McGrath': 1, 'Stefan Ruzowitzky': 1}), 'Harrison For
d': defaultdict(<class 'int'>, {'Steven Spielberg': 4, 'Gavin Hood': 1, 'Robert Z
emeckis': 1, 'Alan J. Pakula': 1, 'Wolfgang Petersen': 1, 'Ivan Reitman': 1, 'Ron
Shelton': 1, 'Sydney Pollack': 1, 'Phillip Noyce': 2, 'Adam McKay': 1, 'Andrew Da
vis': 1, 'Brian Helgeland': 1, 'Richard Marquand': 1, 'Francis Ford Coppola': 2,
'Tom Vaughan': 1, 'Lee Toland Krieger': 1, 'Ridley Scott': 1, 'Irvin Kershner':
1, 'Peter Weir': 1, 'George Lucas': 2, 'Guy Hamilton': 1}), 'A.J. Buckley': defau
ltdict(<class 'int'>, {'Peter Sohn': 1}), 'Kelly Macdonald': defaultdict(<class
'int'>, {'Mark Andrews': 1, 'Joe Wright': 1, 'Ethan Coen': 1, 'Mike Figgis': 1,
'Danny Boyle': 1}), 'Sofia Boutella': defaultdict(<class 'int'>, {'Justin Lin':
1}), 'John Ratzenberger': defaultdict(<class 'int'>, {'Andrew Stanton': 1, 'Pete
Docter': 1, 'John Lasseter': 1}), 'Tzi Ma': defaultdict(<class 'int'>, {'Brett Ra
tner': 1}), 'Oliver Platt': defaultdict(<class 'int'>, {'Roland Emmerich': 1, 'Be
tty Thomas': 1, 'Stuart Baird': 1, 'Gary Fleder': 1, 'Steve Miner': 1, 'Menno Mey
jes': 1, 'Brian Robbins': 1, 'Mark Steven Johnson': 1, 'Nicole Holofcener': 1, 'P
eter Hedges': 1}), 'Robin Wright': defaultdict(<class 'int'>, {'Robert Zemeckis':
2, 'David Fincher': 1, 'M. Night Shyamalan': 1, 'Kevin Macdonald': 1, 'Luis Mando
ki': 1, 'Robert Redford': 1, 'Rob Reiner': 1, 'Michael Mayer': 1, 'Anne Fontain
e': 1}), 'Channing Tatum': defaultdict(<class 'int'>, {'Lana Wachowski': 1, 'Rola
nd Emmerich': 1, 'Jon M. Chu': 1, 'Ron Howard': 1, 'Phil Lord': 2, 'Jorge R. Guti
érrez': 1, 'Thomas Carter': 1, 'Evan Goldberg': 1, 'Michael Sucsy': 1, 'Steven So
derbergh': 3, 'Steve Boyum': 1, 'Andy Fickman': 1, 'Lasse Hallström': 1, 'Dito Mo
ntiel': 1, 'Anne Fletcher': 1}), 'Jim Broadbent': defaultdict(<class 'int'>, {'An
drew Adamson': 1, 'Frank Coraci': 1, 'Sarah Smith': 1, 'Chris Wedge': 1, 'Baz Luh
rmann': 1, 'Gary Chapman': 1, 'Bille August': 1, 'Reinhard Klooss': 1, 'Peter Hew
itt': 1, 'Woody Allen': 1, 'Mike Leigh': 2, 'Sidney J. Furie': 1, 'Ian Fitzgibbo
n': 1, '': 1, 'Mark Herman': 1, 'Neil Jordan': 1}), 'Amy Poehler': defaultdict(<c
lass 'int'>, {'Rob Letterman': 1, 'Pete Docter': 1, 'Mike Mitchell': 2, 'Betty Th
omas': 1, 'Jimmy Hayward': 1, 'Mike Disa': 1, 'Barry Levinson': 1}), 'Chloë Grace
Moretz': defaultdict(<class 'int'>, {'Martin Scorsese': 1, 'Byron Howard': 1, 'Jo
hn Whitesell': 1, 'J Blakeson': 1, 'Nicholas Stoller': 1, 'Kimberly Peirce': 1,
'Jeff Wadlow': 1, 'Matt Reeves': 1, 'Thor Freudenthal': 1, 'David Moreau': 1, 'R.
J. Cutler': 1}), 'Jet Li': defaultdict(<class 'int'>, {'Rob Cohen': 1, 'Richard D
onner': 1, 'Rob Minkoff': 1, 'Yimou Zhang': 1, 'Peter Ho-Sun Chan': 1, 'Andrzej B
artkowiak': 2, 'Chris Nahon': 1}), 'Jimmy Bennett': defaultdict(<class 'int'>,
{'Tom Shadyac': 1, 'Wolfgang Petersen': 1, 'Florent-Emilio Siri': 1, 'Richard Lon
craine': 1, 'Jaume Collet-Serra': 1, 'Robert Rodriguez': 1, 'Andrew Douglas': 1,
'James Mottern': 1}), 'Tom Cruise': defaultdict(<class 'int'>, {'Doug Liman': 1,
'Christopher McQuarrie': 2, 'Brad Bird': 1, 'Edward Zwick': 1, 'Steven Spielber
g': 2, 'John Woo': 1, 'James Mangold': 1, 'Bryan Singer': 1, 'Brian De Palma': 1,
'Cameron Crowe': 2, 'Stanley Kubrick': 1, 'Tony Scott': 2, 'Michael Mann': 1, 'Sy
dney Pollack': 1, 'Rob Reiner': 1, 'Barry Levinson': 1, 'Oliver Stone': 1, 'Marti
n Scorsese': 1, 'Francis Ford Coppola': 1, 'Curtis Hanson': 1}), 'Jeanne Trippleh
orn': defaultdict(<class 'int'>, {'Kevin Reynolds': 1, 'Kelly Makin': 1, 'Peter H
owitt': 1}), 'Joseph Gordon-Levitt': defaultdict(<class 'int'>, {'Stephen Sommer
s': 1, 'Ron Clements': 1, 'Steven Spielberg': 1, 'Frank Miller': 1, 'Spike Lee':
1, 'Brian Gibson': 1, '': 1, 'David Koepp': 1, 'Robert Zemeckis': 1, 'Rian Johnso

```

```

n': 1, 'Kimberly Peirce': 1, 'Gil Junger': 1, 'Jonathan Levine': 1, 'Marc Webb':
1, 'Spencer Susser': 1, 'Joseph Gordon-Levitt': 1, 'C. Jay Cox': 1}), 'Scarlett Johansson': defaultdict(<class 'int'>, {'Jon Favreau': 1, 'Anthony Russo': 1, 'Michael Bay': 1, 'Robert Redford': 1, 'Brian De Palma': 1, 'Frank Miller': 1, 'Cameron Crowe': 1, 'Luc Besson': 1, 'Stephen Hillenborg': 1, 'Ellory Elkayem': 1, 'Paul Weitz': 1, 'Spike Jonze': 1, 'Ethan Coen': 1, 'Woody Allen': 3, 'Jonathan Glazer': 1, 'Peter Webber': 1, 'Terry Zwigoff': 1, 'Sofia Coppola': 1}), 'Angelina Jolie Pitt': defaultdict(<class 'int'>, {'Robert Stromberg': 1, 'Jennifer Yuh Nelson': 1, 'Mark Osborne': 1, 'Phillip Noyce': 1, 'Simon West': 1, 'Timur Bekmambetov': 1, 'Kerry Conran': 1, 'Martin Campbell': 1, 'Clint Eastwood': 1, 'Stephen Herek': 1, 'Iain Softley': 1, 'Mike Newell': 1, 'Michael Cristofer': 1, 'James Mangold': 1, 'Michael Winterbottom': 1}), 'Gary Oldman': defaultdict(<class 'int'>, {'Matt Reeves': 1, 'José Padilha': 1, 'Stephen Hopkins': 1, 'Roland Joffé': 1, 'Catherine Hardwicke': 1, 'Frederik Du Chau': 1, 'Ariel Vromen': 1, 'David S. Goyer': 1, 'Peter Medak': 1, 'Julian Schnabel': 1}), 'Tamsin Egerton': defaultdict(<class 'int'>, {'Roland Joffé': 1}), 'Keanu Reeves': defaultdict(<class 'int'>, {'Carla Rinsch': 1, 'Nancy Meyers': 1, 'Scott Derrickson': 1, 'Lana Wachowski': 1, 'Taylor Hackford': 1, 'Andrew Davis': 1, 'Howard Deutch': 1, 'Francis Ford Coppola': 1, 'Alejandro Agresti': 1, 'Pat O'Connor': 1, 'Joe Charbanic': 1, 'Jan de Bont': 1, 'Stephen Frears': 1, 'Brian Robbins': 1, 'Peter Hewitt': 1, 'David Ayer': 1, 'Stephen Herek': 1, 'Kenneth Branagh': 1, 'Nicolas Winding Refn': 1, 'Stephen Kay': 1, 'Gus Van Sant': 1, 'Tim Hunter': 1}), 'Jon Hamm': defaultdict(<class 'int'>, {'Mike Mitchell': 1, 'Zack Snyder': 1, 'Randall Wallace': 1, 'Craig Gillespie': 1, 'David Wain': 1}), 'Judy Greer': defaultdict(<class 'int'>, {'Brad Bird': 1, 'Peyton Reed': 1, 'Nancy Meyers': 1, 'Mike Nichols': 1, 'David O. Russell': 1, 'Anne Fletcher': 1, 'Brandon Camp': 1, 'Paul Weitz': 1, 'Jay Duplass': 1, 'Chris D'Arienza': 1, 'Adam Goldberg': 1, 'Jonathan Kesselman': 1, 'Craig Mazin': 1}), 'Damon Wayans Jr.': defaultdict(<class 'int'>, {'Don Hall': 1, 'Damien Dante Wayans': 1}), 'Jack McBrayer': defaultdict(<class 'int'>, {'Rich Moore': 1, 'Brad Peyton': 1, 'Ramona Mosley': 1}), 'Vivica A. Fox': defaultdict(<class 'int'>, {'Roland Emmerich': 1, 'Quentin Tarantino': 1, 'Jesse Vaughan': 1, 'Mort Nathan': 1, 'Roldman Flender': 1, 'F. Gary Gray': 1, 'Mark Brown': 1, 'David Boyd': 1}), 'Gerard Butler': defaultdict(<class 'int'>, {'Dean DeBlois': 2, 'Alex Proyas': 1, 'Jan de Bont': 1, 'Antoine Fuqua': 1, 'Zack Snyder': 1, 'Babak Najafi': 1, 'Joel Schumacher': 1, 'F. Gary Gray': 1, 'Mark Neveldine': 1, 'Andy Tennant': 1, 'Robert Luketic': 1, 'Jennifer Flackett': 1, 'Gabriele Muccino': 1, 'Richard LaGravenese': 1, 'Marc Forster': 1, 'Patrick Lussier': 1, 'Michael Apted': 1, 'David Anspaugh': 1, 'Mike Barker': 1, 'Ralph Fiennes': 1, 'Shona Auerbach': 1}), 'Nick Stahl': defaultdict(<class 'int'>, {'Jonathan Mostow': 1, 'Terrence Malick': 1}), 'Bradley Cooper': defaultdict(<class 'int'>, {'James Gunn': 1, 'Todd Phillips': 2, 'Clint Eastwood': 1, 'Garry Marshall': 1, 'Peyton Reed': 1, 'Tom Dey': 1, 'Christian Alvarado': 1, 'David Palmer': 1, 'John Wells': 1, 'Phil Traill': 1, 'Ryûhei Kitamura': 1, 'Dan Trachtenberg': 1}), 'Matthew McConaughey': defaultdict(<class 'int'>, {'Christopher Nolan': 1, 'Breck Eisner': 1, 'Robert Zemeckis': 1, 'Andy Tennant': 1, 'McG': 1, 'Jonathan Mostow': 1, 'Ron Howard': 1, 'Donald Petrie': 1, 'Gary Ross': 1, 'Brad Furman': 1, 'Adam Shankman': 1, 'Richard Linklater': 2, 'Bill Paxton': 1, 'Jeff Nichols': 1, 'William Friedkin': 1, 'S.R. Bindler': 1, 'Jean-Marc Vallée': 1, 'John Sayles': 1, 'Jill Sprecher': 1}), 'Mark Chinnery': defaultdict(<class 'int'>, {'Hideaki Anno': 1}), 'Paul Walker': defaultdict(<class 'int'>, {'Rob Cohen': 2, 'Justin Lin': 1, 'Richard Donner': 1, 'John Singleton': 1, 'Clint Eastwood': 1, 'John Stockwell': 1, 'Frank Marshall': 1, 'John Luessenhop': 1, 'Camille Delamarre': 1, 'Steve Boyum': 1, 'John Dahl': 1, 'Wayne Kramer': 1, 'Brian Robbins': 1, 'Robert Iscove': 1, 'John Herzfeld': 1}), 'Brad Pitt': defaultdict(<class 'int'>, {'David Fincher': 2, 'Wolfgang Petersen': 1, 'Steven Soderbergh': 2, 'Doug Liman': 1, 'Tony Scott': 2, 'David Ayer': 1, 'Jean-Jacques Annaud': 1, 'Patrick Gilmore': 1, 'Neil Jordan': 1, 'Terrence Malick': 1, 'Andrew Dominik': 2, 'Alejandro G. Iñárritu': 1, 'Angelina Jolie Pitt': 1, 'Tom DiCillo': 1}), 'Nicolas Cage': defaultdict(<class 'int'>, {'Jon Turteltaub': 2, 'Mark Steven Johnson': 1, 'John Woo': 2, 'Dominic Sena': 1, 'Michael Bay': 1, 'Brian De Palma': 1, 'Oliver Stone': 1, 'David Bowers': 1, 'Brett Ratner': 1, 'Mark Neveldine': 1, 'Brad Silberl

```

```

ing': 1, 'Alex Proyas': 1, 'Patrick Lussier': 1, 'Francis Ford Coppola': 2, 'John
A. Davis': 1, 'Danny Pang': 1, 'Andrew Niccol': 1, 'Joel Schumacher': 1, 'Barbet
Schroeder': 1, 'Simon West': 1, 'Martin Scorsese': 1, 'Werner Herzog': 1, 'Gore V
erbinski': 1, 'Scott Walker': 1, 'Spike Jonze': 1, 'Vic Armstrong': 1, 'Paul Schr
ader': 1, 'Mike Figgis': 1, 'David Gordon Green': 1}), 'Justin Timberlake': defau
ltdict(<class 'int'>, {'Chris Miller': 1, 'Eric Brevig': 1, 'Marco Schnabel': 1,
'Brad Furman': 1, 'Jake Kasdan': 1, 'Craig Brewer': 1, 'Ethan Coen': 1}), 'Domini
c Cooper': defaultdict(<class 'int'>, {'Duncan Jones': 1, 'Gary Shore': 1, 'Niels
Arden Oplev': 1, 'Saul Dibb': 1, '': 1, 'Lone Scherfig': 1, 'Nicholas Hytner':
1}), 'Bruce Spence': defaultdict(<class 'int'>, {'Michael Apted': 1, 'Steve Oedek
erk': 1}), 'Jennifer Garner': defaultdict(<class 'int'>, {'Michael Bay': 1, 'Pete
r Berg': 1, 'Rob Bowman': 1, 'Peter Hedges': 1, 'Gary Winick': 1, 'Ivan Reitman':
1, 'Ricky Gervais': 1, 'Patricia Riggen': 1, 'Danny Leiner': 1}), 'Zack Ward': de
faultdict(<class 'int'>, {'Michael Bay': 1, 'Zack Ward': 1, 'Marco Kreuzpaintne
r': 1, '': 1, 'Bob Clark': 1}), 'Anthony Hopkins': defaultdict(<class 'int'>, {'O
liver Stone': 2, 'Joe Johnston': 1, 'Darren Aronofsky': 1, 'Martin Brest': 1, 'Le
e Tamahori': 1, 'Joel Schumacher': 1, 'Martin Campbell': 1, 'Jon Turteltaub': 1,
'Steven Spielberg': 1, 'Mikael Håfström': 1, 'Scott Hicks': 1, 'Edward Zwick': 1,
'Roger Donaldson': 1, 'Richard Attenborough': 1, 'Jonathan Demme': 1, 'James Ivor
y': 2, 'Woody Allen': 1, 'Emilio Estevez': 1, 'David Lynch': 1}), 'Robert Pattins
on': defaultdict(<class 'int'>, {'David Yates': 1, 'Mike Newell': 1, 'Bill Condo
n': 1, 'David Slade': 1, 'Chris Weitz': 1, 'Francis Lawrence': 1, 'Allen Coulte
r': 1}), 'Janeane Garofalo': defaultdict(<class 'int'>, {'Brad Bird': 1, 'Kinka U
sher': 1, 'Richard Kelly': 1, 'David Dobkin': 1, 'Risa Bramon Garcia': 1}), 'Bern
ie Mac': defaultdict(<class 'int'>, {'Eric Darnell': 1, 'Malcolm D. Lee': 1, 'Chr
is Rock': 1, 'Kevin Rodney Sullivan': 1, 'Sam Weisman': 1, 'Charles Stone III':
1, 'Terry Zwigoff': 1, 'Lionel C. Martin': 1, 'Ice Cube': 1, 'Spike Lee': 2}), 'R
obin Williams': defaultdict(<class 'int'>, {'Shawn Levy': 3, 'George Miller': 2,
'Chris Columbus': 2, 'Les Mayfield': 1, 'Vincent Ward': 1, 'Steven Spielberg': 1,
'Tom Shadyac': 1, 'Danny DeVito': 1, 'Walt Becker': 1, 'Ken Kwapis': 1, 'Justin Z
ackham': 1, 'Ron Clements': 1, 'Kirsten Sheridan': 1, 'Barry Levinson': 2, 'Peter
Weir': 1, 'Peter Kassovitz': 1, 'Mark Romanek': 1, 'Gus Van Sant': 1, 'Bobcat Gol
dthwait': 1, 'Patrick Stettner': 1, 'David Duchovny': 1, 'James David Pasternak':
1}), 'Essie Davis': defaultdict(<class 'int'>, {'Lana Wachowski': 1, 'Baz Luhrman
n': 1}), 'Josh Gad': defaultdict(<class 'int'>, {'Chris Buck': 1, 'Shawn Levy':
1}), 'Steve Bastoni': defaultdict(<class 'int'>, {'Lana Wachowski': 1}), 'Kelli G
arner': defaultdict(<class 'int'>, {'Hoyt Yeatman': 1, 'Mike Mills': 1, 'Larry Cl
ark': 1}), 'Matthew Broderick': defaultdict(<class 'int'>, {'Steve Hickner': 1,
'Brett Ratner': 1, 'Roger Allers': 1, 'David Kellogg': 1, 'John Whitesell': 1, 'B
en Stiller': 1, 'Jeff Nathanson': 1, 'Richard Donner': 1, 'John Badham': 1, 'Alex
ander Payne': 1, 'Kenneth Lonergan': 1}), 'Seychelle Gabriel': defaultdict(<class
'int'>, {'M. Night Shyamalan': 1}), 'Philip Seymour Hoffman': defaultdict(<class
'int'>, {'J.J. Abrams': 1, 'Jan de Bont': 1, 'Anthony Minghella': 2, 'Brett Ratne
r': 1, 'Mike Nichols': 1, 'Cameron Crowe': 1, 'Bennett Miller': 2, 'Richard Curti
s': 1, 'John Hamburg': 1, 'John Patrick Shanley': 1, 'Charlie Kaufman': 1, 'Anton
Corbijn': 1, 'Joel Coen': 1, 'Richard Kwietniowski': 1, 'Tamara Jenkins': 1, 'Spi
ke Lee': 1, 'Todd Solondz': 1, 'Brad Anderson': 1}), 'Elisabeth Harnois': default
dict(<class 'int'>, {'Simon Wells': 1}), 'Ty Burrell': defaultdict(<class 'int'>,
{'Rob Minkoff': 1, 'Louis Leterrier': 1, 'James Bobin': 1, 'Zack Snyder': 1, 'Cra
ig Johnson': 1}), 'Jada Pinkett Smith': defaultdict(<class 'int'>, {'Eric Darnel
l': 2, 'Tom Shadyac': 1, 'Diane English': 1, 'Keenen Ivory Wayans': 1, 'Matty Ric
h': 1, 'Albert Hughes': 1}), 'Toby Stephens': defaultdict(<class 'int'>, {'Lee Ta
mahori': 1, 'Michael Bay': 1}), 'Ed Begley Jr.': defaultdict(<class 'int'>, {'Pau
l Feig': 1, 'Penny Marshall': 1, 'Ben Stassen': 1, 'Paul Schrader': 2, 'Chris Pai
ne': 1}), 'Bruce Willis': defaultdict(<class 'int'>, {'Michael Bay': 1, 'Len Wise
man': 1, 'John Moore': 1, 'John McTiernan': 1, 'Tim Johnson': 1, 'Dean Parisot':
1, 'Jonathan Mostow': 1, 'Antoine Fuqua': 1, 'Barry Levinson': 1, 'Renny Harlin':
1, 'Gregory Hoblit': 1, 'Jon Turteltaub': 1, 'Michael Lehmann': 1, 'Robert Schwen
tke': 1, 'Harold Becker': 1, 'Robert Zemeckis': 1, 'Rob Reiner': 1, 'Richard Donn
er': 1, 'M. Night Shyamalan': 1, 'Kevin Smith': 1, 'Howard Deutch': 1, 'Paul McGu

```

```

igan': 1, 'Jonathan Lynn': 1, 'Brian A Miller': 1, 'Wes Anderson': 1, 'Mike Judge': 1, 'Alan Rudolph': 1, 'Quentin Tarantino': 1}), 'John Michael Higgins': defaultdict(<class 'int'>, {'Dean Parisot': 1, 'Christopher Guest': 2}), 'Sam Sheppard': defaultdict(<class 'int'>, {'Rob Cohen': 1, 'Mateo Gil': 1, 'Terrence Malick': 1}), 'Matt Frewer': defaultdict(<class 'int'>, {'Zack Snyder': 1, 'Uwe Boll': 1}), 'Kevin Rankin': defaultdict(<class 'int'>, {'Ang Lee': 1}), 'Chris Evans': defaultdict(<class 'int'>, {'Joe Johnston': 1, 'Tim Story': 1, 'Danny Boyle': 1, 'Joon-ho Bong': 1, 'Kevin Munroe': 1, 'Sylvain White': 1, 'Mark Mylod': 1, 'Joel Gallen': 1, 'Ariel Vromen': 1}), 'Colin Salmon': defaultdict(<class 'int'>, {'Michael Apted': 1, '': 1, 'John Simpson': 1, 'Stuart Hazeldine': 1}), 'James D'Arcy': defaultdict(<class 'int'>, {'Peter Weir': 1, 'Renny Harlin': 1}), 'Don Johnson': defaultdict(<class 'int'>, {'': 1, 'Ron Shelton': 1, 'Nick Cassavetes': 1, 'Tom Brady': 1}), 'Mark Rylance': defaultdict(<class 'int'>, {'Steven Spielberg': 1}), 'Matt Damon': defaultdict(<class 'int'>, {'Paul Greengrass': 4, 'Neill Blomkamp': 1, 'Ridley Scott': 1, 'Kelly Asbury': 1, 'Terry Gilliam': 1, 'Don Bluth': 1, 'Doug Liman': 1, 'Steven Soderbergh': 2, 'Clint Eastwood': 2, 'Robert Redford': 1, 'Bobby Farrelly': 1, 'George Nolfi': 1, 'Stephen Gaghan': 1, 'Billy Bob Thornton': 1, 'Francis Ford Coppola': 1, 'Ethan Coen': 1, 'Jeff Schaffer': 1, 'Gus Van Sant': 2, 'Kenneth Lonergan': 1, 'John Dahl': 1, 'Kevin Smith': 2, 'Charles Ferguson': 1}), 'Jim Parsons': defaultdict(<class 'int'>, {'Tim Johnson': 1, 'Zach Braff': 1}), 'Salma Hayek': defaultdict(<class 'int'>, {'Chris Miller': 1, 'Brett Ratner': 1, 'Peter Lord': 1, 'Frank Coraci': 1, 'Joachim Rønning': 1, 'Kevin Hooks': 1, 'Mark Christopher': 1, 'Julie Taymor': 1, 'Allison Anders': 1, 'Dan Ireland': 1}), 'Toby Jones': defaultdict(<class 'int'>, {'Steven Spielberg': 1, 'Ron Howard': 1, 'Oliver Stone': 1, 'John Curran': 1, 'Frank Darabont': 1, 'Douglas McGrath': 1}), 'Daniel Radcliffe': defaultdict(<class 'int'>, {'Alfonso Cuarón': 1, 'Chris Columbus': 2, 'Jon M. Chu': 1, 'Paul McGuigan': 1, 'John Boorman': 1, 'James Watkins': 1}), 'Alfre Woodard': defaultdict(<class 'int'>, {'Eric Leighton': 1, 'Michael Tollin': 1, 'Liz Friedlander': 1, 'Cathy Malkasian': 1, 'Bille Woodruff': 1, 'Gina Prince-Bythewood': 1, 'Spike Lee': 1, 'John R. Leonetti': 1, 'Bill Benenson': 1}), 'Rupert Grint': defaultdict(<class 'int'>, {'Matt Birch': 2, 'Jonathan Lynn': 1, 'Jeremy Brock': 1}), 'Miguel Ferrer': defaultdict(<class 'int'>, {'Carlos Saldanha': 1, 'Les Mayfield': 1, 'John Sayles': 1}), 'Ronny Cox': defaultdict(<class 'int'>, {'Paul Verhoeven': 1}), 'Tony Curran': defaultdict(<class 'int'>, {'John McTiernan': 1, 'Steven Soderbergh': 1, 'Neil Jordan': 1}), 'Jeremy Renner': defaultdict(<class 'int'>, {'Tony Gilroy': 1, 'Clark Johnson': 1, 'Tommy Wirkola': 1, 'Ben Affleck': 1, 'Niki Caro': 1, 'James Gray': 1, 'Juan Carlos Fresnadillo': 1, 'Kathryn Bigelow': 1, 'Michael Cuesta': 1}), 'Michael Gough': defaultdict(<class 'int'>, {'Joel Schumacher': 2, 'Tim Burton': 2}), 'Clint Howard': defaultdict(<class 'int'>, {'Ron Howard': 1, 'Mike Marvin': 1, 'Jay Russell': 1}), 'Karen Allen': defaultdict(<class 'int'>, {'Wolfgang Petersen': 1}), 'Suraj Sharma': defaultdict(<class 'int'>, {'Ang Lee': 1}), 'Demi Moore': defaultdict(<class 'int'>, {'McG': 1, 'Gary Trousdale': 1, 'Barry Levinson': 1, 'Andrew Bergman': 1, 'Jon Cassar': 1, 'Jerry Zucker': 1, 'Derrick Borte': 1}), 'Michael Fassbender': defaultdict(<class 'int'>, {'Ridley Scott': 2, 'Quentin Tarantino': 1, 'Jimmy Hayward': 1, 'David Cronenberg': 1, 'Neil Marshall': 1, 'Steve McQueen': 1, 'John Maclean': 1, 'Andrea Arnold': 1, 'James Watkins': 1}), 'Nathan Lane': defaultdict(<class 'int'>, {'Rob Minkoff': 1, 'Timothy Björklund': 1}), 'Vin Diesel': defaultdict(<class 'int'>, {'David Twohy': 3, 'Breck Eisner': 1, 'Rob Cohen': 1, 'Adam Shankman': 1, 'Brad Bird': 1, 'Mathieu Kassovitz': 1, 'F. Gary Gray': 1, 'Brian Koppelman': 1, 'Ben Younger': 1}), 'Scott Porter': defaultdict(<class 'int'>, {'Lana Wachowski': 1, 'Todd Graff': 1}), 'Shelley Conn': defaultdict(<class 'int'>, {'James L. Brooks': 1}), 'Morgan Freeman': defaultdict(<class 'int'>, {'Joseph Kosinski': 1, 'Timur Bekmambetov': 1, 'Tom Shadyac': 1, 'Mimi Leder': 1, 'Louis Leterrier': 2, 'Mikael Salomon': 1, 'Phil Alden Robinson': 1, 'Lawrence Kasdan': 1, 'Phill Lord': 1, 'George Armitage': 1, 'Rob Reiner': 1, 'Carl Franklin': 1, 'Charles Martin Smith': 2, 'David Fincher': 1, 'Lasse Hallström': 1, 'Lee Tamahori': 1, 'Gary Fleder': 1, 'Frank Darabont': 1, 'Neil LaBute': 1, 'Bruce Beresford': 1, 'Luc Jacquet': 1}), 'Natalie Portman': defaultdict(<class 'int'>, {'George Lucas': 3, 'Tim Burton': 1, 'James McTeigue': 1, 'David Gordon Green': 1, 'Justin Chadwick':

```



```

1, 'Mike Nichols': 1, 'Jim Sheridan': 1, 'Ivan Reitman': 1, "Gavin O'Connor": 1,
'Wayne Wang': 1, 'Matt Williams': 1, 'Darren Aronofsky': 1, 'Kar-Wai Wong': 1}),
'Alain Delon': defaultdict(<class 'int'>, {'Frédéric Forestier': 1}), 'Noel Fish-
er': defaultdict(<class 'int'>, {'Jonathan Liebesman': 2, 'Tim Hill': 1}), 'Phaldu-
t Sharma': defaultdict(<class 'int'>, {'Alfonso Cuarón': 1}), 'Jamie Renée Smit-
h': defaultdict(<class 'int'>, {'Roger Donaldson': 1}), 'Stephen Amell': defaultd-
ict(<class 'int'>, {'Dave Green': 1}), 'Tim Blake Nelson': defaultdict(<class 'in-
t'>, {'Josh Trank': 1, 'Andrew Davis': 1, 'Joey Lauren Adams': 1}), 'Dwayne Johns-
on': defaultdict(<class 'int'>, {'Brad Peyton': 1, 'Adam McKay': 1, 'Brett Ratne-
r': 1, 'Stephen Sommers': 1, 'Peter Berg': 1, 'F. Gary Gray': 1, 'Andrzej Bartkow-
iak': 1, 'Chuck Russell': 1, 'Jorge Blanco': 1, 'Kevin Bray': 1, 'Rawson Marshall
Thurber': 1, 'Andy Fickman': 2, 'Phil Joanou': 1, 'Michael Bay': 1, 'George Tillm-
an Jr.': 1, 'Ric Roman Waugh': 1}), 'Vincent Schiavelli': defaultdict(<class 'in-
t'>, {'Roger Spottiswoode': 1}), 'Heath Ledger': defaultdict(<class 'int'>, {'Rol-
and Emmerich': 1, 'Brian Helgeland': 2, 'Catherine Hardwicke': 1, 'Marc Forster':
1}), 'Kate Winslet': defaultdict(<class 'int'>, {'Robert Schwentke': 1, 'Nancy Me-
yers': 1, 'Neil Burger': 1, 'Steven Zaillian': 1, 'Stephen Daldry': 1, 'Roman Pol-
anski': 1, 'Michel Gondry': 1, 'John Hillcoat': 1, 'Danny Boyle': 1, 'Todd Fiel-
d': 1, 'Philip Kaufman': 1, 'John Turturro': 1, 'Richard Eyre': 1, 'Peter Jackso-
n': 1}), 'James Corden': defaultdict(<class 'int'>, {'Rob Letterman': 1, 'Robert
B. Weide': 1, 'John Carney': 1}), 'George Peppard': defaultdict(<class 'int'>,
{'': 1, 'Jack Smight': 1}), 'Eva Green': defaultdict(<class 'int'>, {'Noam Murr-
o': 1, 'Martin Campbell': 1, 'Benedek Fliegau': 1}), 'Mahadeo Shivraj': defaultd-
ict(<class 'int'>, {'Raja Gosnell': 1}), 'Jacob Tremblay': defaultdict(<class 'in-
t'>, {'Raja Gosnell': 1}), 'Jason Patric': defaultdict(<class 'int'>, {'Jan de Bo-
nt': 1, 'Nick Cassavetes': 1, 'Joe Carnahan': 1}), 'Sean Hayes': defaultdict(<cla-
ss 'int'>, {'Bo Welch': 1}), 'Chazz Palminteri': defaultdict(<class 'int'>, {'Rob
Minkoff': 1, 'Keenen Ivory Wayans': 1, 'Chris Weitz': 1, 'Matthew Diamond': 1, 'P-
aul Mazursky': 1, 'Barrett Esposito': 1, 'Mars Callahan': 1, 'Dave Rodriguez':
1}), 'Oprah Winfrey': defaultdict(<class 'int'>, {'Ron Clements': 1, 'Jonathan De-
mme': 1, 'Steven Spielberg': 1}), 'Mathew Buck': defaultdict(<class 'int'>, {'Chr-
istopher Barnard': 1}), 'Del Zamora': defaultdict(<class 'int'>, {'Peter Chelso-
m': 1, 'Alex Cox': 1}), 'Djimon Hounsou': defaultdict(<class 'int'>, {'Ridley Sco-
tt': 1, 'Stefen Fangmeier': 1, 'Stephen Sommers': 1, 'Shekhar Kapur': 1, 'Jeff Wa-
dlow': 1, 'Julie Taymor': 1, 'Mark Neveldine': 1, 'David E. Talbert': 1}), 'Cary-
Hiroyuki Tagawa': defaultdict(<class 'int'>, {'Tim Burton': 1, 'Simon Wincer': 1,
'Bernardo Bertolucci': 1, 'Lasse Hallström': 1, 'Mark L. Lester': 1}), 'Joe Morto-
n': defaultdict(<class 'int'>, {'James Cameron': 1, 'Tom Shadyac': 1, '': 1}), 'D-
enzel Washington': defaultdict(<class 'int'>, {'Ridley Scott': 1, 'Tony Scott':
4, 'Daniel Espinosa': 1, 'Albert Hughes': 1, 'Jonathan Demme': 2, 'Edward Zwick':
3, 'Baltasar Kormákur': 1, 'Antoine Fuqua': 2, 'Spike Lee': 4, 'Carl Franklin':
1, 'Phillip Noyce': 1, 'Alan J. Pakula': 1, 'Norman Jewison': 1, 'Nick Cassavete-
s': 1, 'Robert Zemeckis': 1, 'Brett Leonard': 1, 'Richard Attenborough': 1, 'Denz-
el Washington': 2}), 'Jamie Lee Curtis': defaultdict(<class 'int'>, {'James Camer-
on': 1, 'Joe Roth': 1, 'Mark Waters': 1, 'Raja Gosnell': 1, 'Andy Fickman': 1, 'J-
ohn Carpenter': 2, 'Rick Rosenthal': 1, 'Amy Holden Jones': 1}), 'Robert De Nir-
o': defaultdict(<class 'int'>, {'Paul Weitz': 1, 'Des McAnuff': 1, 'Bibo Bergero-
n': 1, 'Frank Oz': 1, 'Jay Roach': 2, 'Harold Ramis': 2, 'Jon Avnet': 1, 'John Fr-
ankenheimer': 1, 'Tony Scott': 1, 'Martin Scorsese': 7, 'Barry Levinson': 3, 'Joh-
n Herzfeld': 1, 'Nancy Meyers': 1, 'Michael Caton-Jones': 1, 'Peter Segal': 1, 'S-
cott Mann': 1, 'George Tillman Jr.': 1, 'Martin Brest': 1, 'Nick Hamm': 1, 'Sergi-
o Leone': 1, 'Jon Turteltaub': 1, 'Joel Schumacher': 1, 'Garry Marshall': 1, 'Bri-
an De Palma': 1, 'John Polson': 1, 'Mary McGuckian': 1, 'John Curran': 1, 'Neil J-
ordan': 1, 'Kirk Jones': 1, 'Jonathan Jakubowicz': 1, 'Rodrigo Cortés': 1, 'Micha-
el Cimino': 1, 'James Mangold': 1, 'Terry Gilliam': 1, 'Francis Ford Coppola': 1,
'Ethan Maniquis': 1, 'Quentin Tarantino': 1, 'Dan Mazer': 1}), 'Vanessa William-
s': defaultdict(<class 'int'>, {'Chuck Russell': 1, 'Simon Wincer': 1, 'Gary Halv-
orson': 1, 'Christopher Erskin': 1, 'George Tillman Jr.': 1, 'Richard Schenkman':
1}), 'Eartha Kitt': defaultdict(<class 'int'>, {'Mark Dindal': 1}), "Catherine
O'Hara": defaultdict(<class 'int'>, {'Spike Jonze': 1, 'Mark Dindal': 1, 'Sam Men

```

```

des': 1, 'Tim Burton': 1, 'Dean Parisot': 1, 'Christopher Guest': 1}}, 'Josh Hutcherson': defaultdict(<class 'int'>, {'Chris Wedge': 1, 'Brad Peyton': 1, 'Eric Br evig': 1, 'Jesse Dylan': 1, 'Paul Weitz': 1, 'Gabor Csupo': 1, 'Andrea Di Stefan o': 1, 'Joseph Kahn': 1, 'Lisa Cholodenko': 1, 'Shari Springer Berman': 1}}, 'Jak e Busey': defaultdict(<class 'int'>, {'Paul Verhoeven': 1}}, 'Abbie Cornish': def aultdict(<class 'int'>, {'Zack Snyder': 1, 'Martin McDonagh': 1, 'Jane Campion': 1}}, 'Frances Conroy': defaultdict(<class 'int'>, {'Pitof': 1, 'Anand Tucker': 1, 'Robert Cary': 1}}, 'Will Ferrell': defaultdict(<class 'int'>, {'Brad Silberlin g': 1, 'Jay Roach': 2, 'Adam McKay': 2, 'Josh Gordon': 1, 'Kent Alterman': 1, 'Se an Anders': 1, "Matthew O'Callaghan": 1, 'Etan Cohen': 1, 'Marc Forster': 1, 'Tod d Phillips': 1, 'Kevin Smith': 1, 'John Fortenberry': 1, 'Nick Gomez': 1, 'Bruce McCulloch': 1, 'Reginald Hudlin': 1, 'Andrew Wilson': 1, 'Matt Piedmont': 1, 'Dan Rush': 1}}, 'Ray Winstone': defaultdict(<class 'int'>, {'Ericson Core': 1, 'Pierr e Morel': 1, 'Liliana Cavani': 1, 'John Hillcoat': 1, 'Jason Connery': 1, 'Ian Sh arp': 1, 'Jonathan Glazer': 1, 'Mora Stephens': 1, 'Antonio Simoncini': 1}}, 'Jam ie Kennedy': defaultdict(<class 'int'>, {'Lawrence Guterman': 1}}, 'Rosario Dawso n': defaultdict(<class 'int'>, {'Ron Underwood': 1, 'D.J. Caruso': 1, 'Frank Cora ci': 1, 'Frank Miller': 1, 'Chris Columbus': 1, 'Harry Elfont': 1, 'Danny Boyle': 1, 'Craig Bolotin': 1, 'Chris Rock': 1, 'Kris Isacsson': 1, 'Larry Clark': 1}}, 'Francesca Capaldi': defaultdict(<class 'int'>, {'Steve Martino': 1}}, 'Ben Gazza ra': defaultdict(<class 'int'>, {'David Mamet': 1, 'Rowdy Herrington': 1}}, 'Jon Lovitz': defaultdict(<class 'int'>, {'Frank Oz': 1, 'Demian Lichtenstein': 1, 'Su san Stroman': 1, 'Dennis Dugan': 1, 'Richard Benjamin': 1, 'Gene Quintano': 1, 'J erry Rees': 1}}, 'Ioan Gruffudd': defaultdict(<class 'int'>, {'Ridley Scott': 1, 'Kevin Lima': 1, 'Alister Grierson': 1, 'Jonathan Newman': 1}}, 'Milla Jovovich': defaultdict(<class 'int'>, {'Luc Besson': 1, 'Paul W.S. Anderson': 4, 'Alexander Witt': 1, 'Ben Stiller': 2, 'Russell Mulcahy': 1, 'Kurt Wimmer': 1, 'James McTeig ue': 1, 'Michael Winterbottom': 1, 'William A. Graham': 1, 'Olatunde Osunsanmi': 1, 'Richard Linklater': 1}}, 'Chris Noth': defaultdict(<class 'int'>, {'Michael P atrick King': 1, '': 1, 'Daniel Sackheim': 1, 'Michael Radford': 1}}, 'Frank Welk er': defaultdict(<class 'int'>, {'Bibo Bergeron': 1, 'Luis Llosa': 1, 'Maurice Jo yce': 1}}, 'Hayley Atwell': defaultdict(<class 'int'>, {'Kenneth Branagh': 1, 'Ni ck Love': 1}}, 'Michael Imperioli': defaultdict(<class 'int'>, {'Peter Jackson': 1, 'Brad Anderson': 1, 'Cheryl Dunye': 1, 'Spike Lee': 1}}, 'Alexander Gould': de faultdict(<class 'int'>, {'Andrew Stanton': 1, 'Robert Harmon': 1}}, 'Orlando Blo om': defaultdict(<class 'int'>, {'Peter Jackson': 1, 'Cameron Crowe': 1, 'J  r  me Salle': 1}}, 'Michael Jeter': defaultdict(<class 'int'>, {'Joe Johnston': 1, 'Ant hony Russo': 1}}, 'James Franco': defaultdict(<class 'int'>, {'Rupert Wyatt': 1, 'Ryan Murphy': 1, 'Tony Bill': 1, 'John Dahl': 1, 'Evan Goldberg': 1, 'Greg Tiern an': 1, 'David Gordon Green': 1, 'Paul Haggis': 1, 'Gus Van Sant': 1, 'Danny Boyl e': 1, 'David Raynr': 1, 'Harmony Korine': 1, 'Karen Moncrieff': 1, 'Gia Coppol a': 1}}, 'Martin Short': defaultdict(<class 'int'>, {'Mark Waters': 1, 'Brenda Ch apman': 1, 'John A. Davis': 1, 'John Blanchard': 1}}, 'Dennis Quaid': defaultdict (<class 'int'>, {'John Lee Hancock': 2, 'John Moore': 1, 'Lawrence Kasdan': 1, 'R ob Cohen': 1, 'Pete Travis': 1, 'Gary Fleder': 1, 'Gregory Hoblit': 1, 'Philip Ka ufman': 1, 'Scott Stewart': 1, 'Sean McNamara': 1, 'Todd Haynes': 1, 'Walter Hil l': 1}}, 'Holly Hunter': defaultdict(<class 'int'>, {'Brad Bird': 1, 'Nick Hurra n': 1, 'Joel Coen': 1, 'Daniel Barnz': 1, 'Richard LaGravenese': 1, 'Jane Campio n': 1, '': 1, 'Catherine Hardwicke': 1}}, 'Christopher Masterson': defaultdict(<c lass 'int'>, {'Renny Harlin': 1}}, 'Logan Lerman': defaultdict(<class 'int'>, {'C hris Columbus': 1, 'Thor Freudenthal': 1, 'Joel Schumacher': 1, 'James Schamus': 1, 'Wil Shriner': 1, 'Eric Bress': 1, 'Stephen Chbosky': 1}}, 'Mei Melan  on': def aultdict(<class 'int'>, {'Brett Ratner': 1}}, 'Will Forte': defaultdict(<class 'i nt'>, {'Phil Lord': 1, 'Cody Cameron': 1, 'Akiva Schaffer': 1}}, 'Denis Leary': d efaultdict(<class 'int'>, {'Carlos Saldanha': 2}}, 'Adam Scott': defaultdict(<cla ss 'int'>, {'Ben Stiller': 1, 'Robert Luketic': 1, 'Alexandre Aja': 1, 'Anand Tuc ker': 1, 'Michael Dougherty': 1, 'Steve Pink': 1, 'Richard Shepard': 1, 'Nicholas Fackler': 1}}, 'Bill Murray': defaultdict(<class 'int'>, {'McG': 1, 'Peter Sega l': 1, 'Joe Pytko': 1, 'George Clooney': 1, 'Tim Hill': 1, 'Peter Hewitt': 1, 'We s Anderson': 5, 'Bobby Farrelly': 2, 'Gil Kenan': 1, 'Richard Donner': 1, 'Tim Ro

```

```

bbins': 1, 'Frank Oz': 1, 'John McNaughton': 1, 'Jon Amiel': 1, 'Sydney Pollack':
1, 'Harold Ramis': 2, 'Theodore Melfi': 1, 'Ivan Reitman': 1, 'Don Kempf': 1, 'Aa
ron Schneider': 1}), 'Ming-Na Wen': defaultdict(<class 'int'>, {'Tony Bancroft':
1, 'Steven E. de Souza': 1, 'Nelson McCormick': 1}), 'Leonard Nimoy': defaultdict
(<class 'int'>, {'Gary Trousdale': 1, 'Robert Wise': 1, 'William Shatner': 1, 'Ni
cholas Meyer': 2, 'Leonard Nimoy': 2, 'Wayne Thornley': 1}), 'Bella Thorne': defa
ultdict(<class 'int'>, {'Walt Becker': 1, 'Frank Coraci': 1, 'Miguel Arteta': 1,
'Ari Sandel': 1, 'Tyler Oliver': 1}), 'Adam Sandler': defaultdict(<class 'int'>,
{'Dennis Dugan': 3, 'Frank Coraci': 2, 'Peter Segal': 2, 'Adam Shankman': 1, 'Ste
ven Brill': 1, 'James L. Brooks': 1, 'Judd Apatow': 1, 'Sean Anders': 1, 'Paul Th
omas Anderson': 1, 'Mike Binder': 1}), 'Haley Joel Osment': defaultdict(<class 'i
nt'>, {'Steven Spielberg': 1, 'Norman Jewison': 1, 'Tim McCanlies': 1, 'Steve Tre
nbirth': 1, 'Peter Hastings': 1}), 'Marsha Thomason': defaultdict(<class 'int'>,
{'Rob Minkoff': 1}), 'Greg Grunberg': defaultdict(<class 'int'>, {'Paul Verhoeve
n': 1, 'John Whitesell': 1}), 'Curtiss Cook': defaultdict(<class 'int'>, {'Sydney
Pollack': 1}), 'Alun Armstrong': defaultdict(<class 'int'>, {'Phillip Noyce':
1}), 'Don Cheadle': defaultdict(<class 'int'>, {'Brian De Palma': 1, 'Mick Jackso
n': 1, 'Thor Freudenthal': 1, 'Steven Soderbergh': 1, 'Terry George': 1, 'Antoine
Fuqua': 1, 'Paul Thomas Anderson': 1, 'Paul Haggis': 1, 'John Michael McDonagh':
1}), 'Anne Hathaway': defaultdict(<class 'int'>, {'Carlos Saldanha': 1, 'Garry Ma
rshall': 2, "Tommy O'Haver": 1, 'Gary Winick': 1, 'Cory Edwards': 1, 'Julian Jarr
old': 1, 'Lone Scherfig': 1, 'Jonathan Demme': 1, 'Mitch Davis': 1, 'Kate Barker-
Froyland': 1, '': 1}), 'William Smith': defaultdict(<class 'int'>, {'John Miliu
s': 1, 'Scott Ziehl': 1}), 'Paddy Considine': defaultdict(<class 'int'>, {'Ron Ho
ward': 1, 'Jamie Thraves': 1, 'Pawel Pawlikowski': 1, 'Shane Meadows': 1}), 'Shir
ley Henderson': defaultdict(<class 'int'>, {'Andrey Konchalovskiy': 1, 'Sally Pot
ter': 1, '': 1}), 'Jeff Bennett': defaultdict(<class 'int'>, {'Kevin Lima': 1}),
'Caroline Dhavernas': defaultdict(<class 'int'>, {'': 1, 'Josh Gordon': 1}), 'Mer
yl Streep': defaultdict(<class 'int'>, {'Nancy Meyers': 1, 'Curtis Hanson': 1, 'N
ora Ephron': 1, 'David Frankel': 2, 'Robert Redford': 1, 'Sydney Pollack': 1, 'Ca
rl Franklin': 1, 'Stephen Frears': 1, 'Stephen Daldry': 1, 'Phyllida Lloyd': 1,
'Robert Altman': 1, 'Fred Zinnemann': 1}), 'Al Pacino': defaultdict(<class 'in
t'>, {'Steven Soderbergh': 1, 'Dennis Dugan': 1, 'Michael Mann': 1, 'Oliver Ston
e': 1, 'Francis Ford Coppola': 2, 'Christopher Nolan': 1, 'Harold Becker': 2, 'Jo
n Avnet': 1, 'Michael Radford': 1, 'Brian De Palma': 1, 'Daniel Algrant': 1, 'Shi
ntaro Shimomura': 1, 'Dan Fogelman': 1}), 'Jon Favreau': defaultdict(<class 'in
t'>, {'Roger Allers': 1, 'Seth Gordon': 2, 'Peter Billingsley': 1, 'Peyton Reed':
1, 'Peter Berg': 1, 'Hart Bochner': 1, 'Jon Favreau': 1, 'Doug Liman': 1}), 'Bob
Hoskins': defaultdict(<class 'int'>, {'Jean-Jacques Annaud': 1, 'Wayne Wang': 1,
'Annabel Jankel': 1, 'Stephen Frears': 1, 'Atom Egoyan': 1, 'Fred Schepisi': 1,
'Nigel Cole': 1, 'Sacha Bennett': 1, 'Shane Meadows': 2}), 'F. Murray Abraham': d
efaultdict(<class 'int'>, {'John McTiernan': 1, 'Gus Van Sant': 1}), 'Li Gong': d
efaultdict(<class 'int'>, {'Rob Marshall': 1, 'Pou-Soi Cheang': 1, 'Peter Webbe
r': 1, 'Yimou Zhang': 2, 'Kar-Wai Wong': 1}), 'Amber Stevens West': defaultdict(<
class 'int'>, {'Justin Lin': 1}), 'Raymond Cruz': defaultdict(<class 'int'>, {'An
drew Davis': 1}), 'Roy Scheider': defaultdict(<class 'int'>, {'Bob Fosse': 1, 'Wi
lliam Friedkin': 2, 'Jeannot Szwarc': 1, 'Steven Spielberg': 1}), 'Julia Robert
s': defaultdict(<class 'int'>, {'Tarsem Singh': 1, 'Richard Donner': 1, 'Garry Ma
rshall': 2, 'Mike Newell': 1, 'Tony Gilroy': 1, 'Steven Soderbergh': 2, 'Chris Co
lumbus': 1, 'Stephen Frears': 1, 'P.J. Hogan': 1, 'Joe Roth': 1, 'Roger Michell':
1, 'Jodie Foster': 1, 'Joel Schumacher': 1, 'Billy Ray': 1, 'Donald Petrie': 1}),
'Anna Kendrick': defaultdict(<class 'int'>, {'Edgar Wright': 1, 'Chris Butler':
1, 'Kirk Jones': 1, 'Elizabeth Banks': 1, 'Jason Moore': 1, 'Richard LaGravenes
e': 1, 'Joe Swanberg': 1}), 'Larry Miller': defaultdict(<class 'int'>, {'Peter Se
gal': 1, 'James Dodson': 1}), 'Sarah Michelle Gellar': defaultdict(<class 'int'>,
{'Raja Gosnell': 2, 'Paul Bolger': 1, 'Takashi Shimizu': 2, 'Roger Kumble': 1, 'E
mily Young': 1, '': 1, 'Mark Tarlov': 1, 'James Toback': 1}), 'Wood Harris': defa
ultdict(<class 'int'>, {'Pete Travis': 1}), 'Ted Danson': defaultdict(<class 'in
t'>, {'George A. Romero': 1, 'Ken Kwapis': 1}), 'Kristen Stewart': defaultdict(<c
lass 'int'>, {'Doug Liman': 1, 'Jon Favreau': 1, 'David Fincher': 1, 'Catherine H

```

```
ardwicke': 1, 'Walter Salles': 1, 'Sean Penn': 1, 'Bart Freundlich': 1, 'Udayan P
rasad': 1, 'Jon Kasdan': 1, 'Jake Scott': 1, 'Greg Mottola': 1, 'Floria Sigismond
i': 1, 'Richard Glatzer': 1}}, 'Seth MacFarlane': defaultdict(<class 'int'>, {'Gu
illermo del Toro': 1}), 'Robert Duvall': defaultdict(<class 'int'>, {'Roger Spott
iswoode': 2, 'Steven Zaillian': 1, 'Jon Turteltaub': 1, 'James Gray': 1, 'Kevin C
ostner': 1, 'Billy Bob Thornton': 1, 'Sidney Lumet': 1, 'Robert Mulligan': 1}),
'Tia Carrere': defaultdict(<class 'int'>, {'Dean DeBlois': 1, 'Penelope Spheer
i': 1, 'Myles Berkowitz': 1}), 'Elden Henson': defaultdict(<class 'int'>, {'': 1,
'Troy Miller': 1}), 'Roger Rees': defaultdict(<class 'int'>, {'Shawn Levy': 1, 'O
liver Hirschbiegel': 1, 'Robin Budd': 1}), 'Vanessa Redgrave': defaultdict(<class
'int'>, {'Henry Jaglom': 1, 'Gary Winick': 1, 'Roland Emmerich': 1, 'James Ivor
y': 1}), 'Elizabeth Montgomery': defaultdict(<class 'int'>, {'': 1}), 'Quincy Jon
es': defaultdict(<class 'int'>, {'James Algar': 1}), 'Charlize Theron': defaulttdi
ct(<class 'int'>, {'Ron Underwood': 1, 'Karyn Kusama': 1, 'John Frankenheimer':
1, 'James Gray': 1, 'Jason Reitman': 1, 'Patty Jenkins': 1}), 'Michael Emerson':
defaultdict(<class 'int'>, {'Martin Campbell': 1, 'Jay Oliva': 1, 'James Wan':
1}), 'Nissim Renard': defaultdict(<class 'int'>, {'Lucile Hadzihalilovic': 1}),
'Bruce Greenwood': defaultdict(<class 'int'>, {'Roger Donaldson': 1, 'Bruce Beres
ford': 2, 'Frederik Du Chau': 1, 'David Nutter': 1, 'Mark Piznarski': 1, 'Guy Rit
chie': 1, 'Atom Egoyan': 2, 'Kelly Reichardt': 1, 'Andrew Niccol': 1}), 'Sylveste
r Stallone': defaultdict(<class 'int'>, {'Rob Cohen': 1, 'Renny Harlin': 2, 'Mika
el Häfström': 1, 'Eric Darnell': 1, 'Peter MacDonald': 1, 'Andrey Konchalovskiy':
1, 'Jim Gillespie': 1, 'Walter Hill': 1, 'Ted Kotcheff': 1, 'Luis Llosa': 1, 'Geo
rge P. Cosmatos': 1, 'Robert Rodriguez': 1, 'Stephen Kay': 1, 'Ryan Coogler': 1,
'Sylvester Stallone': 1, 'Norman Jewison': 1, 'Bruce Malmuth': 1, 'John G. Avilds
en': 1, 'Paul Bartel': 1}), 'Charlie Rowe': defaultdict(<class 'int'>, {'Barry Co
ok': 1}), 'Richard Tyson': defaultdict(<class 'int'>, {'Roger Christian': 1, 'Iva
n Reitman': 1, 'Terron R. Parsons': 1}), 'Brendan Fraser': defaultdict(<class 'in
t'>, {'Joe Dante': 1, 'Henry Selick': 1, 'Iain Softley': 1, 'Sam Weisman': 1, 'Ha
rold Ramis': 1, 'Cal Bruner': 1, 'Hugh Wilson': 2, 'Roger Kumble': 1, 'Richard B
enjamin': 1, 'Phillip Noyce': 1, 'Bill Condon': 1}), 'Fergie': defaultdict(<class
'int'>, {'Rob Marshall': 1}), 'Olivia Williams': defaultdict(<class 'int'>, {'Kev
in Costner': 1, 'Thaddeus O'Sullivan': 1}), 'Adam Goldberg': defaultdict(<class
'int'>, {'George Miller': 1, 'Ron Howard': 1, 'Bibo Bergeron': 1, 'William Brent
Bell': 1, 'D.J. Caruso': 1}), 'Bob Neill': defaultdict(<class 'int'>, {'Antony Ho
ffman': 1}), 'Mia Farrow': defaultdict(<class 'int'>, {'Luc Besson': 1, 'Nadia Ta
ss': 1}), 'Pedro Armendáriz Jr.': defaultdict(<class 'int'>, {'Jacques Perrin':
1}), 'David Oyelowo': defaultdict(<class 'int'>, {'Peter Hyams': 1, 'Anthony Hemi
ngway': 1, 'Ava DuVernay': 1, 'J.C. Chandor': 1, 'Kevin Macdonald': 1, 'Jerry Jam
eson': 1}), 'Sasha Roiz': defaultdict(<class 'int'>, {'Paul W.S. Anderson': 1}),
'Sariann Monaco': defaultdict(<class 'int'>, {'Andrés Couturier': 1}), 'Steve Coo
gan': defaultdict(<class 'int'>, {'Yarrow Cheney': 1, 'Frank Oz': 1, 'Stephen Fre
ars': 1, 'Andrew Fleming': 1, 'Michael Winterbottom': 1}), 'Jason Flemyng': defau
ltdict(<class 'int'>, {'Stephen Norrington': 1, '': 1, 'Alexandre Aja': 1, 'Don M
ancini': 1, 'Oleg Stepchenko': 1, 'Bob Spiers': 1, 'Jonathan English': 1}), 'Stev
e Carell': defaultdict(<class 'int'>, {'Pierre Coffin': 2, 'Kyle Balda': 1, 'Jay
Roach': 1, 'Judd Apatow': 1, 'Joe Nussbaum': 1, 'Lorene Scafaria': 1, 'Jonathan D
ayton': 1, 'Peter Sollett': 1, 'Nat Faxon': 1}), 'Ariana Richards': defaultdict(<
class 'int'>, {'Steven Spielberg': 1, 'Patrick Read Johnson': 1}), 'Jerry Stille
r': defaultdict(<class 'int'>, {'Adam Shankman': 1, 'Lawrence Kasanoff': 1, 'Eric
Bross': 1}), 'Harold Perrineau': defaultdict(<class 'int'>, {'': 1, 'Bob Rafelso
n': 1, 'Malcolm D. Lee': 2, 'Fina Torres': 1}), 'Nicholas Lea': defaultdict(<clas
s 'int'>, {'Martin Campbell': 1}), 'Loretta Devine': defaultdict(<class 'int'>,
{'Bill Condon': 1, 'John Ottman': 1, 'Michael Schultz': 1}), 'Adam Arkin': defaul
tdict(<class 'int'>, {'': 1, 'Diane Keaton': 1}), 'Ayelet Zurer': defaultdict(<cl
ass 'int'>, {'Steven Spielberg': 1}), 'Tom Selleck': defaultdict(<class 'int'>,
{'Robert Luketic': 1, 'Frank Oz': 1, 'Leonard Nimoy': 1}), 'Julia Ormond': defaul
tdict(<class 'int'>, {'Jerry Zucker': 1, 'Patricia Rozema': 1}), 'Bai Ling': defa
ultdict(<class 'int'>, {'Andy Tennant': 1}), 'Connie Nielsen': defaultdict(<class
'int'>, {'Paul W.S. Anderson': 1, 'William Friedkin': 1, 'John McTiernan': 1, 'Mc
```

```
G': 1, 'Vic Sarin': 1}}, 'Christopher Meloni': defaultdict(<class 'int'>, {'David
Pastor': 1, 'Lana Wachowski': 1, 'David Wain': 1}), 'Peter Coyote': defaultdict(<
class 'int'>, {'Barry Levinson': 1, 'Brian De Palma': 1, 'Michael Polish': 1}),
'Raven-Symoné': defaultdict(<class 'int'>, {'Steve Carr': 1}), 'Mhairi Calvey': d
efaultdict(<class 'int'>, {'Mel Gibson': 1}), 'Albert Brooks': defaultdict(<class
'int'>, {'David Silverman': 1, 'Joe Dante': 1}), 'Martin Landau': defaultdict(<cl
ass 'int'>, {'Frank Darabont': 1, 'Rob Bowman': 1, 'Joseph L. Mankiewicz': 1, 'Ti
m Burton': 1, 'Shane Acker': 1, 'Steve Barron': 1, 'George Stevens': 1, 'Ronald N
eame': 1, 'Daniel Davila': 1}), 'David Gant': defaultdict(<class 'int'>, {'Jean-J
acques Annaud': 1}), 'Rory Culkin': defaultdict(<class 'int'>, {'M. Night Shyamal
an': 1, 'Arie Posin': 1, 'Jacob Aaron Estes': 1}), 'Rupert Everett': defaultdict
(<class 'int'>, {'Andrew Adamson': 1}), 'Tim Conway': defaultdict(<class 'int'>,
{'Paul Tibbitt': 1, '': 1}), 'Lili Taylor': defaultdict(<class 'int'>, {'Ron Howa
rd': 1}), 'Nicole Beharie': defaultdict(<class 'int'>, {'': 1, 'Stefan C. Schaeffe
r': 1}), 'Sanaa Lathan': defaultdict(<class 'int'>, {'Paul W.S. Anderson': 1, 'St
ephenn Norrington': 1}), 'LeVar Burton': defaultdict(<class 'int'>, {'Jonathan Fra
kes': 2, 'David Carson': 1, '': 1}), 'David Hyde Pierce': defaultdict(<class 'in
t'>, {'Mike Nichols': 1, 'Nick Tomnay': 1}), 'Jeremy Irvine': defaultdict(<class
'int'>, {'Steven Spielberg': 1, 'Jonathan Teplitzky': 1, 'Roland Emmerich': 1, 'O
l Parker': 1}), 'Michael Biehn': defaultdict(<class 'int'>, {'James Cameron': 3,
'William Friedkin': 1, 'Christian Duguay': 1, 'Jonathan Frakes': 1, 'George P. Co
smatos': 1, 'Brian Trenchard-Smith': 1, 'Xavier Gens': 1}), 'Frank Langella': def
aultdict(<class 'int'>, {'Oliver Stone': 1, 'Richard Kelly': 1, 'Franklin J. Scha
ffner': 1, 'Jake Schreier': 1}), 'Gary Cole': defaultdict(<class 'int'>, {'Betty
Thomas': 1, 'Tim Hill': 1, 'Rawson Marshall Thurber': 1, 'Ben Falcone': 1, 'Sam R
aimi': 1, 'Mike Judge': 1, 'Mitch Davis': 1}), 'Theo James': defaultdict(<class
'int'>, {'Måns Mårland': 1}), 'James Martin Kelly': defaultdict(<class 'int'>,
{'Adam Shankman': 1}), 'LL Cool J': defaultdict(<class 'int'>, {'John McTiernan':
1, 'Renny Harlin': 2, 'Wayne Wang': 1, 'Daisy von Scherler Mayer': 1, 'Michael Ry
mer': 1}), 'Talisa Soto': defaultdict(<class 'int'>, {'Wych Kaosayananda': 1}),
'Ron Howard': defaultdict(<class 'int'>, {'Bobby Farrelly': 1}), 'Lea Michele': d
efaultdict(<class 'int'>, {'Will Finn': 1, 'Kevin Tancharoen': 1}), 'Charlotte Ra
mpling': defaultdict(<class 'int'>, {'Michael Caton-Jones': 1, 'Norman Jewison':
1, 'Sidney Lumet': 1, 'François Ozon': 1, 'Hans Petter Moland': 1, 'Jean-Jacques
Mantello': 1}), 'Roxanne McKee': defaultdict(<class 'int'>, {'Renny Harlin': 1}),
'Vincent Pastore': defaultdict(<class 'int'>, {'Joseph Ruben': 1, 'Rob Pritts':
1, 'James Melkonian': 1, 'Danny Provenzano': 1}), 'Max Minghella': defaultdict(<c
lass 'int'>, {'Alejandro Amenábar': 1, 'Chris Gorak': 1}), 'Jenna Fischer': defau
ltdict(<class 'int'>, {'Bobby Farrelly': 1, 'Brian Koppelman': 1, 'Bob Odenkirk':
1}), 'Michael Raymond-James': defaultdict(<class 'int'>, {'Craig Gillespie': 1}),
'Rufus Sewell': defaultdict(<class 'int'>, {'Timur Bekmambetov': 1, 'Chuck Russel
l': 1, 'Christian Duguay': 1, 'Alex Proyas': 1, 'Neil Burger': 1, 'Edward Hall':
1}), 'Will Patton': defaultdict(<class 'int'>, {'Jon Amiel': 1}), 'Hector Elizond
o': defaultdict(<class 'int'>, {'': 1, 'Robert Butler': 1, 'Garry Marshall': 1,
'Jesse Dylan': 1}), 'Colin Firth': defaultdict(<class 'int'>, {'Doug Lefler': 1,
'Phyllida Lloyd': 1, 'Beeban Kidron': 1, 'Richard Curtis': 1, 'Anthony Minghell
a': 1, 'Sharon Maguire': 1, 'Kirk Jones': 1, 'Dennie Gordon': 1, 'Atom Egoyan':
1, 'Rowan Joffe': 1, 'Tom Hooper': 1, 'Oliver Parker': 1, 'Stephan Elliott': 1,
'Anand Tucker': 1, 'Tom Ford': 1}), 'Rami Malek': defaultdict(<class 'int'>, {'Sc
ott Waugh': 1, 'Spike Lee': 1}), 'Goran Visnjic': defaultdict(<class 'int'>, {'Ch
ris Wedge': 1, 'Griffin Dunne': 1, 'Mike Mills': 1}), 'Judge Reinhold': defaulttdi
ct(<class 'int'>, {'Michael Lembeck': 1, 'John Pasquin': 1, 'Martin Brest': 1}),
'Finn Wittrock': defaultdict(<class 'int'>, {'Angelina Jolie Pitt': 1}), 'Clint E
astwood': defaultdict(<class 'int'>, {'Clint Eastwood': 10, 'Robert Lorenz': 1,
'Joshua Logan': 1, 'Don Siegel': 1, 'Ted Post': 1, 'Sergio Leone': 2}), 'Delroy L
indo': defaultdict(<class 'int'>, {'John Woo': 1, 'Taylor Hackford': 1, 'Barry So
nnenfeld': 1, 'David Webb Peoples': 1}), 'Sayed Badreya': defaultdict(<class 'in
t'>, {'Larry Charles': 1}), 'Quvenzhané Wallis': defaultdict(<class 'int'>, {'Wil
l Gluck': 1, 'Steve McQueen': 1, 'Benh Zeitlin': 1}), 'Lea Thompson': defaultdict
(<class 'int'>, {'John Milius': 1, 'Robert Zemeckis': 3, 'Willard Huyck': 1, 'Ale
```

```

x Ranarivelo': 1}}, 'Maura Tierney': defaultdict(<class 'int'>, {'Mike Nichols': 1, 'Tom Shadyac': 1}), 'Melina Kanakaredes': defaultdict(<class 'int'>, {'Renny Harlin': 1}), 'Pamela Reed': defaultdict(<class 'int'>, {'Taylor Hackford': 1}), 'Michael Rapaport': defaultdict(<class 'int'>, {'Nora Ephron': 1, 'Paul Feig': 1, 'Marc Bennett': 1, 'Hal Haberman': 1}), 'Caitlin Stasey': defaultdict(<class 'int'>, {'Stuart Beattie': 1}), 'Ian McNeice': defaultdict(<class 'int'>, {'Roman Polanski': 1}), 'Michael Kelly': defaultdict(<class 'int'>, {'Baltasar Kormákur': 1, 'Marc Lawrence': 1, 'Josh Trank': 1, 'Peter Stebbings': 1}), 'Michael Smiley': defaultdict(<class 'int'>, {'Tom Tykwer': 1, 'Edgar Wright': 1, 'Ben Wheatley': 1}), 'Verne Troyer': defaultdict(<class 'int'>, {'Jay Roach': 1}), 'Wayne Knight': defaultdict(<class 'int'>, {'Steven Spielberg': 1}), 'Si Won Choi': defaultdict(<class 'int'>, {'Daniel Lee': 1}), 'Shawna Waldron': defaultdict(<class 'int'>, {'Rob Reiner': 1}), 'Craig Stark': defaultdict(<class 'int'>, {'Quentin Tarantino': 1}), 'Eric Dane': defaultdict(<class 'int'>, {'David Frankel': 1, 'Steve Antin': 1, 'Rich Cowan': 1}), 'Jill Hennessy': defaultdict(<class 'int'>, {'Walt Becker': 1}), 'Patrick Fugit': defaultdict(<class 'int'>, {'David Fincher': 1, 'Peter Kosminsky': 1, 'Jonas Åkerlund': 1}), 'Izabella Scorupco': defaultdict(<class 'int'>, {'Martin Campbell': 1}), 'Daniel von Baren': defaultdict(<class 'int'>, {'Simon West': 1}), 'Natascha McElhone': defaultdict(<class 'int'>, {'Peter Weir': 1, 'William Malone': 1}), 'Anjelica Huston': defaultdict(<class 'int'>, {'Steve Carr': 1, 'Barry Sonnenfeld': 1, 'Andy Tennant': 1, 'Bob Rafelson': 1, 'Mira Nair': 1, 'Stewart Raffill': 1, 'Clark Gregg': 1, 'Vincent Gallo': 1}), 'Carol Ann Susi': defaultdict(<class 'int'>, {'Lawrence Guterman': 1}), 'Dorian Missick': defaultdict(<class 'int'>, {'Marc Lawrence': 1, '': 1, 'Qasim Basir': 1}), 'Taylour Lautner': defaultdict(<class 'int'>, {'Adam Shankman': 1, 'Frank Coraci': 1, 'Robert Rodriguez': 1}), 'Ki Hong Lee': defaultdict(<class 'int'>, {'Wes Ball': 2}), 'Billy Burke': defaultdict(<class 'int'>, {'Jay Russell': 1, 'Gregory Hoblit': 1, 'Robert Towne': 1, 'Charles Matthau': 1, 'David F. Sandberg': 1, '': 1}), 'Misha Gabriel Hamilton': defaultdict(<class 'int'>, {'Kenny Ortega': 1}), 'Estella Warren': defaultdict(<class 'int'>, {'David McNally': 1, 'Wayne Kramer': 1}), 'Jennifer Saunders': defaultdict(<class 'int'>, {'Henry Selick': 1}), 'Zooey Deschanel': defaultdict(<class 'int'>, {'M. Night Shyamalan': 1, 'Garth Jennings': 1, 'Barry Sonnenfeld': 1, 'Lawrence Kasdan': 1, 'Ed Decter': 1, 'Michael Clancy': 1, 'Jesse Peretz': 1, 'Adam Rapp': 1, 'David Gordon Green': 1}), 'Snoop Dogg': defaultdict(<class 'int'>, {'Todd Phillips': 1, 'Alan Jacobs': 1}), 'Jim Belushi': defaultdict(<class 'int'>, {'Brian Levant': 1, 'Roman Polanski': 1, 'John Pasquin': 1, 'Oliver Stone': 1}), 'James Babson': defaultdict(<class 'int'>, {'Guillermo del Toro': 1}), 'Ivana Milicevic': defaultdict(<class 'int'>, {'John Woo': 1, 'Charles Robert Carner': 1, 'Jeff Franklin': 1}), 'Emma Watson': defaultdict(<class 'int'>, {'Sam Fell': 1, 'Alejandro Amenábar': 1}), 'Bianca Kajlich': defaultdict(<class 'int'>, {'': 1, 'Ruben Fleischer': 1}), 'Romany Malco': defaultdict(<class 'int'>, {'Kevin Donovan': 1, 'Tim Story': 1}), 'Peter Greene': defaultdict(<class 'int'>, {'Geoff Murphy': 1, 'Les Mayfield': 1, 'Chuck Russell': 1, 'Ash Baron-Cohen': 1}), 'Colm Feore': defaultdict(<class 'int'>, {'Kenneth Branagh': 1, 'Rob Marshall': 1, 'Kasi Lemmons': 1, 'Erik Canuel': 1}), 'Gary Dourdan': defaultdict(<class 'int'>, {'Jean-Pierre Jeunet': 1, 'Gary Fleder': 1, 'Salim Akil': 1}), 'Tate Donovan': defaultdict(<class 'int'>, {'Antoine Fuqua': 1}), 'Isaac Hempstead Wright': defaultdict(<class 'int'>, {'Graham Annable': 1, 'Nick Murphy': 1}), 'Eileen Brennan': defaultdict(<class 'int'>, {'John Pasquin': 1, 'Howard Zieff': 1, 'Victor Salva': 1, 'George Roy Hill': 1}), 'Ryan Gosling': defaultdict(<class 'int'>, {'Ruben Fleischer': 1, 'Glenn Ficarra': 1, 'Barbet Schroeder': 1, 'Boaz Yakin': 1, 'Nick Cassavetes': 1, 'Adam McKay': 1, 'George Clooney': 1, 'Andrew Jarecki': 1, 'Nicolas Winding Refn': 2, 'Derek Cianfrance': 2, 'Craig Gillespie': 1, 'Henry Bean': 1, 'Ryan Fleck': 1, 'Alex Smith': 1}), 'Olivia Wilde': defaultdict(<class 'int'>, {'Harold Ramis': 1, 'Paul Haggis': 1, 'Jessie Nelson': 1, 'David Gelb': 1, 'Joe Swanberg': 1, 'Hans Canosa': 1}), 'Steven Anthony Lawrence': defaultdict(<class 'int'>, {'Donald Petrie': 1}), 'Blair Brown': defaultdict(<class 'int'>, {'Clark Johnson': 1}), 'Cedric the Entertainer': defaultdict(<class 'int'>, {'Joel Coen': 1, 'Darnell Martin': 1}), 'Bojana Novakovic': defaultdict(<class 'int'>, {'Martin Campbell': 1, 'Sam Raimi': 1, 'John Erick Dowdle': 1}),

```

```

'John Kapelos': defaultdict(<class 'int'>, {'Peter Hyams': 1}), 'Sunny Mabrey': d
eaultdict(<class 'int'>, {'Lee Tamahori': 1}), 'Josh Wingate': defaultdict(<clas
s 'int'>, {'Scott Stewart': 1}), 'Sean Pertwee': defaultdict(<class 'int'>, {'Pau
l W.S. Anderson': 1, 'Danny Cannon': 1, 'Martyn Pick': 1, 'Gary Sinyor': 1}), 'Cl
ifton Collins Jr.': defaultdict(<class 'int'>, {'Rod Lurie': 1, 'William Dear':
1, 'Christine Jeffs': 1, 'Benni Diez': 1, 'Brandon Trost': 1}), 'Robert Forster':
defaultdict(<class 'int'>, {'Walter Hill': 1, 'Bobby Farrelly': 1, 'Hyung-rae Shi
m': 1, 'Gary Nelson': 1}), 'Matt Bomer': defaultdict(<class 'int'>, {'Akiva Golds
man': 1, 'Andrew Niccol': 1, 'Jonathan Liebesman': 1, 'Gregory Jacobs': 1}), 'Mar
c Blucas': defaultdict(<class 'int'>, {'Brian Robbins': 1, 'Michael Tollin': 1,
'Robby Henson': 1}), 'Dougray Scott': defaultdict(<class 'int'>, {'Walter Salle
s': 1}), 'Kelly Preston': defaultdict(<class 'int'>, {'Stephen Herek': 1, 'Steve
Oedekerk': 1, 'Julie Anne Robinson': 1, 'Ivan Reitman': 1}), 'Odeya Rush': defaul
tdict(<class 'int'>, {'Rob Letterman': 1}), 'Nate Richert': defaultdict(<class 'i
nt'>, {'': 1}), 'Jon Heder': defaultdict(<class 'int'>, {'Mark Waters': 1, 'Todd
Phillips': 1, 'Lijun Sun': 1, 'Jared Hess': 1}), 'Taylor Negron': defaultdict(<cl
ass 'int'>, {'Brian Levant': 1, 'Amy Heckerling': 1, 'Alex Zamm': 1, 'Lance Kawa
s': 1}), 'Robert Baker': defaultdict(<class 'int'>, {'George Clooney': 1}), 'Barr
y Corbin': defaultdict(<class 'int'>, {'': 1, 'Tommy Lee Jones': 1, 'Steve Rash':
1, 'Robert M. Young': 1, 'Kirk Loudon': 1}), 'Jesse McCartney': defaultdict(<clas
s 'int'>, {'Tim Hill': 1, 'Bradley Parker': 1}), 'Marlon Brando': defaultdict(<cl
ass 'int'>, {'Richard Donner': 1, 'John Frankenheimer': 1, 'Elia Kazan': 2}), 'Al
yson Hannigan': defaultdict(<class 'int'>, {'Jesse Dylan': 1, 'Jon Hurwitz': 1,
'J.B. Rogers': 1, 'Aaron Seltzer': 1, 'Robert Iscove': 1, 'Alan Cohn': 1, 'Paul W
eitz': 1, 'Bob Saget': 1}), 'Mila Kunis': defaultdict(<class 'int'>, {'Shawn Lev
y': 1, 'Seth MacFarlane': 1, 'Will Gluck': 1, 'John Moore': 1, 'Nicholas Stolle
r': 1, 'Guillaume Canet': 1, 'Jon Lucas': 1, "Tommy O'Haver": 1}), 'Eric Idle': d
efaultdict(<class 'int'>, {'Brad Silberling': 1, 'Marc F. Adler': 1, 'Arthur Hill
er': 1, 'Terry Gilliam': 1}), 'Erika Christensen': defaultdict(<class 'int'>, {'R
obert Schwentke': 1, 'Bob Dolman': 1, 'John Polson': 1}), 'David Carradine': defa
ultdict(<class 'int'>, {'Quentin Tarantino': 1, 'Jason Friedberg': 1, 'Steve Carv
er': 1}), 'Christopher Judge': defaultdict(<class 'int'>, {'': 1, 'David Hewlet
t': 1}), 'Xander Berkeley': defaultdict(<class 'int'>, {'Tom Dey': 1}), 'Madeline
Carroll': defaultdict(<class 'int'>, {'Mark Waters': 1, 'Brian Levant': 1, 'Joshua
Michael Stern': 1, 'Simon West': 1, 'Rob Reiner': 1}), 'Charles S. Dutton': def
aultdict(<class 'int'>, {'David Fincher': 1}), 'Andrea Corr': defaultdict(<class
'int'>, {'Alan Parker': 1}), 'Tom Wilkinson': defaultdict(<class 'int'>, {'Stephe
n Hopkins': 1, 'Gil Junger': 1, 'Ken Scott': 1, 'John Madden': 3, 'Tony Gilroy':
1, 'Alejandro Monteverde': 1, 'Scott Derrickson': 1, 'Gillian Armstrong': 1, 'Pet
er Cattaneo': 1, 'Todd Field': 1}), 'Julie Walters': defaultdict(<class 'int'>,
{'Paul King': 1, 'John Crowley': 1, 'Stephen Daldry': 1}), 'Eddie Redmayne': defa
ultdict(<class 'int'>, {'Shekhar Kapur': 1, 'James Marsh': 1, 'Simon Curtis': 1,
'Tom Kalin': 1}), 'Billy Campbell': defaultdict(<class 'int'>, {'Ron Maxwell':
1}), 'Stephen Root': defaultdict(<class 'int'>, {'Karey Kirkpatrick': 1, 'Kevin S
mith': 1, 'Ridley Scott': 1, 'Callie Khouri': 1, 'Jeff Lowell': 1}), 'Sophia Myle
s': defaultdict(<class 'int'>, {'Jonathan Frakes': 1, 'Len Wiseman': 2}), 'Denis
Lavant': defaultdict(<class 'int'>, {'Jean-Pierre Jeunet': 1, 'Christophe Ali':
1}), 'James Mason': defaultdict(<class 'int'>, {'Stanley Kubrick': 1, 'Anthony Ma
nn': 1, 'Richard Fleischer': 1}), 'Norman Reedus': defaultdict(<class 'int'>, {'G
uillermo del Toro': 1, 'Christian Alvart': 1, 'Davis Guggenheim': 1, 'Scott Kalve
rt': 1, 'Troy Duffy': 2}), 'Todd Stashwick': defaultdict(<class 'int'>, {'Anthony
Russo': 1}), 'Margot Kidder': defaultdict(<class 'int'>, {'Richard Lester': 2}),
'Todd Giebenhain': defaultdict(<class 'int'>, {'Martin Brest': 1}), 'Kirsten Duns
t': defaultdict(<class 'int'>, {'Don Bluth': 1, 'Joe Dante': 1, 'Sofia Coppola':
2, 'Richard Loncraine': 1, 'Jeff Nichols': 1, 'John Stockwell': 1, 'Peyton Reed':
1, 'Michael Patrick Jann': 1, '': 1, 'Leslye Headland': 1}), 'Quentin Tarantino':
defaultdict(<class 'int'>, {'Robert Rodriguez': 3, 'Morgan Spurlock': 1, 'Quentin
Tarantino': 1}), 'Jennifer Ehle': defaultdict(<class 'int'>, {'Kathryn Bigelow':
1, 'Sam Taylor-Johnson': 1, "Gavin O'Connor": 1, 'Mike Binder': 1, 'Shana Feste':
1}), 'Alice Greczyn': defaultdict(<class 'int'>, {'Jay Chandrasekhar': 1}), 'Matt

```



```

Price': defaultdict(<class 'int'>, {'Milos Forman': 1}), 'Kasia Smutniak': default
dict(<class 'int'>, {'Pierre Morel': 1}), 'Jaime King': defaultdict(<class 'in
t'>, {'Paul Hunter': 1, 'Dewey Nicks': 1}), 'Rob Paulsen': defaultdict(<class 'in
t'>, {'Steve Oedekerk': 1, 'Guillaume Ivernel': 1}), 'Kathleen Freeman': defaultd
ict(<class 'int'>, {'Andrew Adamson': 1}), 'Lupe Ontiveros': defaultdict(<class
'int'>, {'James L. Brooks': 1, 'Miguel Arteta': 1}), 'Joel Courtney': defaultdict
(<class 'int'>, {'J.J. Abrams': 1, '': 1}), 'Bill Duke': defaultdict(<class 'in
t'>, {'Brian Helgeland': 1, 'Mark L. Lester': 1, 'Craig R. Baxley': 1}), 'Dylan W
alsh': defaultdict(<class 'int'>, {'Frank Marshall': 1}), 'Fann Wong': defaultdic
t(<class 'int'>, {'David Dobkin': 1}), 'Bernard Hill': defaultdict(<class 'int'>,
{'': 1}), 'Scoot McNairy': defaultdict(<class 'int'>, {'Angela Robinson': 1, 'Gar
eth Edwards': 1}), 'Olivier Martinez': defaultdict(<class 'int'>, {'Adrian Lyne':
1, 'Jean-Paul Rappeneau': 1, 'Katja von Garnier': 1, 'Bigas Luna': 1}), 'Alex Pet
tyfer': defaultdict(<class 'int'>, {'D.J. Caruso': 1, 'Geoffrey Sax': 1, 'Lee Dan
iels': 1, 'Shana Feste': 1, 'Daniel Barnz': 1}), 'Viggo Mortensen': defaultdict(<
class 'int'>, {'Ridley Scott': 1, 'David Cronenberg': 2, 'Agustín Díaz Yanes': 1,
'John Hillcoat': 1, 'Ed Harris': 1, 'Vicente Amorim': 1, 'Tony Goldwyn': 1, 'Davi
d Oelhoffen': 1, 'Kevin Spacey': 1, 'Renny Harlin': 1, 'Gregory Widen': 1}), 'Deb
orah Kara Unger': defaultdict(<class 'int'>, {'David Fincher': 1, 'Geoffrey Sax':
1}), 'Radha Mitchell': defaultdict(<class 'int'>, {'Christophe Gans': 1, 'Michael
J. Bassett': 1, 'Roger Spottiswoode': 1, 'Breck Eisner': 1, 'Joel Schumacher': 1,
'Marc Forster': 1, 'Emma-Kate Croghan': 1}), 'Matt Walsh': defaultdict(<class 'in
t'>, {'Steven Quale': 1, 'Jay Duplass': 1}), 'Louis Lombardi': defaultdict(<class
'int'>, {'John Landis': 1, 'Luke Greenfield': 1, 'James Foley': 1}), 'Armin Muell
er-Stahl': defaultdict(<class 'int'>, {'Mimi Leder': 1}), 'Mike Starr': defaulttdi
ct(<class 'int'>, {'Steven Seagal': 1, 'Woody Allen': 1, 'Michael Schultz': 1, 'B
rian Caunter': 1}), 'Amber Valletta': defaultdict(<class 'int'>, {'Garry Marshal
l': 1}), 'Shô Kosugi': defaultdict(<class 'int'>, {'James McTeigue': 1}), 'David
Jason': defaultdict(<class 'int'>, {'': 1}), 'Muse Watson': defaultdict(<class 'i
nt'>, {'Richard Donner': 1, 'Jay Roach': 1, 'Danny Cannon': 1, 'Jim Gillespie':
1, 'Ed Gass-Donnelly': 1}), 'J.D. Evermore': defaultdict(<class 'int'>, {'Andrew
Niccol': 1}), 'Grey Damon': defaultdict(<class 'int'>, {'': 1}), 'Christina Milia
n': defaultdict(<class 'int'>, {'Stephen Herek': 1}), 'Paul Brooke': defaultdict
(<class 'int'>, {'Luc Besson': 1}), 'Mia Kirshner': defaultdict(<class 'int'>,
{'Costa-Gavras': 1}), 'Julie Benz': defaultdict(<class 'int'>, {'': 1, 'Lexi Alex
ander': 1, 'David Hackl': 1, 'Gary Hardwick': 1, 'Darren Stein': 1}), 'Ian Zierin
g': defaultdict(<class 'int'>, {'Tony Scott': 1, 'Anthony C. Ferrante': 1}), 'Ald
en Ehrenreich': defaultdict(<class 'int'>, {'Richard LaGravenese': 1, 'Francis Fo
rd Coppola': 1}), 'Chris Klein': defaultdict(<class 'int'>, {'Andrzej Bartkowia
k': 1}), 'Tobias Menzies': defaultdict(<class 'int'>, {'': 1}), 'Lauren Holly': d
efaultdict(<class 'int'>, {'Renny Harlin': 1, 'Peter Farrelly': 1}), "Michael O'N
eill": defaultdict(<class 'int'>, {'Steven Soderbergh': 1}), 'P.J. Soles': default
dict(<class 'int'>, {'Chuck Bowman': 1}), 'Corinna Harney': defaultdict(<class
'int'>, {'Jerry Zucker': 1}), 'Nandita Das': defaultdict(<class 'int'>, {'Deepa M
ehta': 1}), 'Charlie Korsmo': defaultdict(<class 'int'>, {'Warren Beatty': 1}),
'Mark Margolis': defaultdict(<class 'int'>, {'John McTiernan': 1, 'Matt Reeves':
1, 'Darren Aronofsky': 2}), 'Rosie Perez': defaultdict(<class 'int'>, {'Penny Mar
shall': 1, 'Patricia Cardoso': 1}), 'Donatas Banionis': defaultdict(<class 'in
t'>, {'Andrei Tarkovsky': 1}), 'Joseph Gilgun': defaultdict(<class 'int'>, {'Brad
Furman': 1, 'James Mather': 1, 'Daniel Barber': 1}), 'Laurence Olivier': defaultd
ict(<class 'int'>, {'Terence Young': 1, 'Franklin J. Schaffner': 1, 'Alfred Hitch
cock': 1}), 'Jonathan Winters': defaultdict(<class 'int'>, {'Brian Levant': 1, 'R
ussell Mulcahy': 1, 'Stanley Kramer': 1}), 'Regina Hall': defaultdict(<class 'in
t'>, {'David Zucker': 1}), 'Candice Bergen': defaultdict(<class 'int'>, {'Donald
Petrie': 1, 'Richard Attenborough': 1}), 'Bradley Whitford': defaultdict(<class
'int'>, {'Joel Schumacher': 1}), 'Natasha Lyonne': defaultdict(<class 'int'>, {'K
eenen Ivory Wayans': 1, 'Woody Allen': 1, 'Adam Rifkin': 1, 'Tamara Jenkins': 1,
'Danny Perez': 1, 'Jamie Babbit': 1}), 'Bill Cobbs': defaultdict(<class 'int'>,
{'James Bobin': 1, 'Michael Apted': 1, 'Tony Kaye': 1, 'Forest Whitaker': 1, 'Mic
k Jackson': 1, 'Michael O. Sajbel': 1}), 'Sydney Pollack': defaultdict(<class 'in

```

```

t'}, {'Roger Michell': 1}), 'Adam Garcia': defaultdict(<class 'int'>, {'David McNally': 1, 'Sara Sugarman': 1, 'Klaus Menzel': 1}), 'Lucille Ball': defaultdict(<class 'int'>, {'Melville Shavelson': 1}), 'Dania Ramirez': defaultdict(<class 'int'>, {'Joel Zwick': 1}), 'Demián Bichir': defaultdict(<class 'int'>, {'Oliver Stone': 1, 'Chris Weitz': 1}), 'Richard Burgi': defaultdict(<class 'int'>, {'David R. Ellis': 1}), 'Kevin McNally': defaultdict(<class 'int'>, {'Peter Howitt': 1, 'Irwin Winkler': 1}), 'Virginia Madsen': defaultdict(<class 'int'>, {'David Lynch': 1, 'Alexander Payne': 1, 'Bobby Roth': 1}), 'Jim Sturgess': defaultdict(<class 'int'>, {'Julie Taymor': 1, 'Giuseppe Tornatore': 1, 'Kari Skogland': 1}), 'Alexa PenaVega': defaultdict(<class 'int'>, {'John Gray': 1, 'Ulu Grosbard': 1, 'Rob Reiner': 1, 'Robert Rodriguez': 3, 'Darren Lynn Bousman': 1, 'Mark Young': 1, 'John Gunn': 1}), 'John de Lancie': defaultdict(<class 'int'>, {'Harold Ramis': 1, 'Marc Schölermann': 1}), 'Malese Jow': defaultdict(<class 'int'>, {'John Schultz': 1}), 'Costas Mandylor': defaultdict(<class 'int'>, {'Sean Penn': 1, 'Kevin Greutert': 2, 'Darren Lynn Bousman': 2}), 'Marcela Mar': defaultdict(<class 'int'>, {'Mike Newell': 1}), 'Pierre Perrier': defaultdict(<class 'int'>, {'': 1}), 'Sam Waterson': defaultdict(<class 'int'>, {'George P. Cosmatos': 1}), 'Daniel Kaluuya': defaultdict(<class 'int'>, {'Oliver Parker': 1}), 'Clea DuVall': defaultdict(<class 'int'>, {'Ben Affleck': 1, 'James Mangold': 1}), 'Mel Blanc': defaultdict(<class 'int'>, {'Norman Ferguson': 1}), 'Judith Chapman': defaultdict(<class 'int'>, {'Roger Kumble': 1}), 'Augustus Prew': defaultdict(<class 'int'>, {'Burr Steers': 1}), 'Imelda Staunton': defaultdict(<class 'int'>, {'Peter Lord': 1, 'Ang Lee': 1}), 'John Witherspoon': defaultdict(<class 'int'>, {'Reginald Hudlin': 1, 'Wes Craven': 1, 'Marcus Raboy': 1, 'Steve Carr': 1}), 'Debra Messing': defaultdict(<class 'int'>, {'Mark Pellington': 1, 'Edward Burns': 1}), 'Bono': defaultdict(<class 'int'>, {'Larry Charles': 1, 'Catherine Owens': 1}), 'Robert Davi': defaultdict(<class 'int'>, {'John Glen': 1}), 'Fisher Stevens': defaultdict(<class 'int'>, {'Julie Anne Robinson': 1, 'Frédéric Auburtin': 1, 'Walter Hill': 1, 'Meiert Avis': 1, 'Avi Nesher': 1}), 'Tony Cox': defaultdict(<class 'int'>, {'Sngmoo Lee': 1}), 'Omar Sy': defaultdict(<class 'int'>, {'Jean-Pierre Jeunet': 1}), 'Mekhi Phifer': defaultdict(<class 'int'>, {'Curtis Hanson': 1, 'Wayne Beach': 1, 'Preston A. Whitmore II': 1, 'Tim Blake Nelson': 1}), 'Jamel Debbouze': defaultdict(<class 'int'>, {'Jamel Debbouze': 1}), 'Julian Sands': defaultdict(<class 'int'>, {'Gordon Chan': 1, 'Frank Marshall': 1, 'Steve Miner': 1, 'James Ivory': 1, 'Anthony Hickox': 1}), 'Robert Clohessy': defaultdict(<class 'int'>, {'Asger Leth': 1}), 'Joel McHale': defaultdict(<class 'int'>, {'David Frankel': 1}), 'Martin Kove': defaultdict(<class 'int'>, {'John G. Avildsen': 1, 'Mario Van Peebles': 1}), 'Tom Welling': defaultdict(<class 'int'>, {'Shawn Levy': 1}), 'Jerry Ferrara': defaultdict(<class 'int'>, {'Peter Berg': 1, 'Michael Corrente': 1}), 'Javier Botet': defaultdict(<class 'int'>, {'James Wan': 1, 'Andrés Muschietti': 1, 'Brian Dorton': 1}), 'Andrew Garfield': defaultdict(<class 'int'>, {'David Fincher': 1, 'Terry Gilliam': 1, 'Mark Romanek': 1, 'Julian Jarrold': 1}), 'Carmen Perez': defaultdict(<class 'int'>, {'Ken Kwapis': 1}), 'Beau Mirchoff': defaultdict(<class 'int'>, {'David Zucker': 1}), 'Kelly Rutherford': defaultdict(<class 'int'>, {'Wes Craven': 1}), 'Olivia Munn': defaultdict(<class 'int'>, {'Tim Story': 1, 'Scott Derrickson': 1, 'Douglas McGrath': 1}), 'Wes Studi': defaultdict(<class 'int'>, {'Michael Mann': 1}), 'Bokeem Woodbine': defaultdict(<class 'int'>, {'Taylor Hackford': 1, 'Kirk Wong': 1, 'Matthew Bright': 1, 'Christian Sesma': 1, 'Tommy Oliver': 1}), 'Jason Alexander': defaultdict(<class 'int'>, {'Bobby Farrelly': 1}), 'Sally Kirkland': defaultdict(<class 'int'>, {'Oliver Stone': 1, 'Sharon Greytak': 1}), 'Matthew Perry': defaultdict(<class 'int'>, {'Burr Steers': 1, 'Reginald Hudlin': 1, 'Damon Santostefano': 1}), 'Krista Allen': defaultdict(<class 'int'>, {'David R. Ellis': 1, 'John Gulager': 1}), 'Joaquim de Almeida': defaultdict(<class 'int'>, {'John Moore': 1, 'James Manera': 1}), 'Pink': defaultdict(<class 'int'>, {'Nicholas Stoller': 1}), 'Lisa Ann Walter': defaultdict(<class 'int'>, {'Peter Chelsom': 1, 'Steven Brill': 1}), 'Michael Jai White': defaultdict(<class 'int'>, {'Mark A.Z. Dippé': 1, 'Mic Rodgers': 1, 'Andrzej Bartkowiak': 1, 'Tyler Perry': 2, 'Ekachai Uekrongtham': 1, 'Ernie Barbarash': 1, 'Kevin Carraway': 1, 'Michael Jai White': 1}), 'Rudy Youngblood': defaultdict(<class 'int'>, {'Mel Gibson': 1, 'Russell Friedenberg': 1}), 'Joe Don Baker': defaultdict(<class 'int'>, {'John Gl

```

```

en': 1}}, 'Topher Grace': defaultdict(<class 'int'>, {'Nimród Antal': 1, 'Robert
Luketic': 1, 'Michael Dowse': 1, 'Jason Stone': 1}), 'Brian Dennehy': defaultdict
(<class 'int'>, {'Ivan Reitman': 1, 'Jean-François Richet': 1, 'Robert Moresco':
1, 'Pete Jones': 1}), 'Gena Rowlands': defaultdict(<class 'int'>, {'Iain Softle
y': 1}), 'Busy Philipps': defaultdict(<class 'int'>, {'Paul Weiland': 1, 'Keenen
Ivory Wayans': 1, 'Joel Edgerton': 1}), 'Johnny Cannizzaro': defaultdict(<class
'int'>, {'Clint Eastwood': 1}), 'Lorraine Bracco': defaultdict(<class 'int'>, {'J
ohn McTiernan': 1}), 'Sam Trammell': defaultdict(<class 'int'>, {'Colin Strause':
1, 'Chuck Russell': 1}), 'Macaulay Culkin': defaultdict(<class 'int'>, {'Donald P
etrie': 1, 'Chris Columbus': 2, 'Howard Zieff': 1, 'Brian Dannelly': 1, 'Fenton B
ailey': 1}), 'Bobby Lee': defaultdict(<class 'int'>, {'Greg Mottola': 1}), 'Miria
m Margolyes': defaultdict(<class 'int'>, {'Anne Fletcher': 1, 'Chris Noonan': 1,
'István Szabó': 1, 'Barbra Streisand': 1}), 'Alison Brie': defaultdict(<class 'in
t'>, {'Wes Craven': 1, 'Christian Ditter': 1, 'Nicholas Stoller': 1}), 'Michael W
incott': defaultdict(<class 'int'>, {'Oliver Stone': 2, 'Alex Proyas': 1}), 'Jame
s Wilcox': defaultdict(<class 'int'>, {'Jake Kasdan': 1}), 'Emma Bell': defaulttdi
ct(<class 'int'>, {'Steven Quale': 1, 'Davis Guggenheim': 1}), 'Jimmy Fallon': de
faultdict(<class 'int'>, {'Bobby Farrelly': 1, 'Dave Borthwick': 1}), 'Bruce Dins
more': defaultdict(<class 'int'>, {'': 1}), 'Noah Bean': defaultdict(<class 'in
t'>, {'Roger Michell': 1}), '50 Cent': defaultdict(<class 'int'>, {'Jim Sherida
n': 1}), 'Oliver Hudson': defaultdict(<class 'int'>, {'Sam Weisman': 1, 'Glen Mor
gan': 1}), 'Catherine Deneuve': defaultdict(<class 'int'>, {'Peter Hyams': 1, 'La
rs von Trier': 1, 'André Téchiné': 1, 'François Ozon': 1, 'Vincent Paronnaud': 1,
'Dick Richards': 1, 'François Truffaut': 1}), 'Stephen Rea': defaultdict(<class
'int'>, {'Stephen Hopkins': 1, 'Neil Jordan': 1, 'Lluís Quílez': 1}), 'Jordi Moll
à': defaultdict(<class 'int'>, {'Olivier Megaton': 1, 'Roland Joffé': 1}), 'Mira
Sorvino': defaultdict(<class 'int'>, {'Irwin Winkler': 1, 'Antoine Fuqua': 1, 'Sp
ike Lee': 1, 'Christopher M. Bessette': 1, 'Griffin Dunne': 1}), 'Dane Cook': def
aultdict(<class 'int'>, {'Joseph Kahn': 1, 'Mark Helfrich': 1, 'Howard Deutch':
1, 'Greg Coolidge': 1}), 'Bobbie Phillips': defaultdict(<class 'int'>, {'Paul Ver
hoeven': 1}), 'Carol Kane': defaultdict(<class 'int'>, {'Elaine May': 1}), 'Richa
rd Epcar': defaultdict(<class 'int'>, {'John Carpenter': 1}), 'Mathieu Kassovit
z': defaultdict(<class 'int'>, {'Jean-Pierre Jeunet': 1}), 'Mary-Kate Olsen': def
aultdict(<class 'int'>, {'Dennie Gordon': 1, 'Jonathan Levine': 1}), 'Omar Epps':
defaultdict(<class 'int'>, {'Charles Shyer': 1, 'Charles S. Dutton': 1, 'Wes Crav
en': 1, 'Takeshi Kitano': 1, 'Rick Famuyiwa': 1}), 'Justin Theroux': defaultdict
(<class 'int'>, {'Danny DeVito': 1, 'David Wain': 1, 'Greg Berlanti': 1}), 'M. Em
met Walsh': defaultdict(<class 'int'>, {'Jerry Jameson': 1}), 'Bridgette Wilson-S
ampras': defaultdict(<class 'int'>, {'Jean-Marie Poiré': 1}), 'John Gatins': defa
ultdict(<class 'int'>, {'Brian Robbins': 1}), 'Sarah Roemer': defaultdict(<class
'int'>, {'D.J. Caruso': 1}), 'James Brolin': defaultdict(<class 'int'>, {'Richard
Shepard': 1}), 'Jennifer Jason Leigh': defaultdict(<class 'int'>, {'Joel Coen':
1, 'John Maybury': 1, 'David Cronenberg': 1, 'Dan Curtis': 1, 'Jane Campion': 1,
'Tuck Tucker': 1, 'Duke Johnson': 1, 'Amy Heckerling': 1}), 'Jay Hernandez': defa
ultdict(<class 'int'>, {'Sergey Bodrov': 1, 'Boaz Yakin': 1, 'Neil LaBute': 1, 'E
li Roth': 2}), 'Tamannaah Bhatia': defaultdict(<class 'int'>, {'S.S. Rajamouli':
1}), 'Arliss Howard': defaultdict(<class 'int'>, {'Robert Schwentke': 1}), 'Colin
Ferguson': defaultdict(<class 'int'>, {'Michael Lehmann': 1}), 'Adam Baldwin': de
faultdict(<class 'int'>, {'Joss Whedon': 1, 'Stephen Hopkins': 1, 'Richard Donne
r': 1, 'Robert Redford': 1}), 'Takeshi Kaneshiro': defaultdict(<class 'int'>, {'J
ohn Woo': 1, 'Yimou Zhang': 1}), 'Ethan Embry': defaultdict(<class 'int'>, {'Andy
Tennant': 1, 'Harry Elfont': 1, 'Danny Leiner': 1, 'Tim McCanlies': 1}), 'Austin
Pendleton': defaultdict(<class 'int'>, {'Jonathan Lynn': 1, 'Billy Wilder': 1}),
'Tony Chiu Wai Leung': defaultdict(<class 'int'>, {'Kar-Wai Wong': 1}), 'Blythe A
uffarth': defaultdict(<class 'int'>, {'Yuefeng Song': 1}), 'Sarah Silverman': def
aultdict(<class 'int'>, {'David Mirkin': 1, 'Bobby Farrelly': 1, 'Scott Alexande
r': 1}), 'D.B. Woodside': defaultdict(<class 'int'>, {'Andy Fickman': 1}), 'Sonia
Braga': defaultdict(<class 'int'>, {'Luis Mandoki': 1}), 'Beth Grant': defaultdic
t(<class 'int'>, {'Stephen Herek': 1, 'Jason Bateman': 1}), 'Mo'Nique': defaulttdi
ct(<class 'int'>, {'Lee Daniels': 1, 'Malcolm D. Lee': 1, 'John Singleton': 1, 'J

```

```

essy Terrero': 1, 'Mark Brown': 1, 'DJ Pooh': 1, 'Nnegest Likké': 1}}, 'Wilford B
rimley': defaultdict(<class 'int'>, {'John Carpenter': 1, 'Peter Segal': 1, 'Rich
ard Dutcher': 1}), 'Gregory Scott Cummins': defaultdict(<class 'int'>, {'Jeb Stua
rt': 1}), 'Tom Skerritt': defaultdict(<class 'int'>, {'Steve Miner': 1, 'Dominic
Sena': 1, 'David Cronenberg': 1, 'Gary Sherman': 1, 'Ridley Scott': 1, 'Mario Van
Peebles': 1}), 'Mike Howard': defaultdict(<class 'int'>, {'Paul Thomas Anderson':
1}), 'Hayden Christensen': defaultdict(<class 'int'>, {'David Leland': 1, 'Sarik
Andreasyan': 1, 'Joby Harold': 1, 'George Hickenlooper': 1, 'Billy Ray': 1, 'Mich
ael Polish': 1}), 'Patton Oswalt': defaultdict(<class 'int'>, {'Paul Thomas Ander
son': 1, 'Todd Strauss-Schulson': 1}), 'Laura-Leigh': defaultdict(<class 'int'>,
{'Rawson Marshall Thurber': 1}), 'Yuliya Snigir': defaultdict(<class 'int'>, {'Fe
dor Bondarchuk': 1}), 'Kathleen Quinlan': defaultdict(<class 'int'>, {'Jonathan M
ostow': 1}), 'Bernie Casey': defaultdict(<class 'int'>, {'Irvin Kershner': 1}),
'Charlie McDermott': defaultdict(<class 'int'>, {'Steve Pink': 1, 'Courtney Hun
t': 1}), 'Rick Yune': defaultdict(<class 'int'>, {'Scott Hicks': 1, 'RZA': 1}),
'Blair Underwood': defaultdict(<class 'int'>, {'Andrew Niccol': 1, 'Michael Schul
tz': 1}), 'Stockard Channing': defaultdict(<class 'int'>, {'Andrew Bergman': 1,
'Gary David Goldberg': 1, 'Hugh Wilson': 1, 'Patrick Stettner': 1}), 'Cheryl Hine
s': defaultdict(<class 'int'>, {'Kirk De Micco': 1}), 'Mike Tyson': defaultdict(<
class 'int'>, {'Wilson Yip': 1}), 'Rene Russo': defaultdict(<class 'int'>, {'Rich
ard Donner': 1}), 'Catherine Dyer': defaultdict(<class 'int'>, {'John Lee Hancoc
k': 1}), 'Jon Foo': defaultdict(<class 'int'>, {'': 1, 'Prachya Pinkaew': 1}), 'A
my Schumer': defaultdict(<class 'int'>, {'Judd Apatow': 1}), 'Michael Angarano':
defaultdict(<class 'int'>, {'Mike Mitchell': 1, 'Thomas Vinterberg': 1, 'Gavin Wi
esen': 1, 'David Gordon Green': 1}), 'Natasha Henstridge': defaultdict(<class 'in
t'>, {'Roger Donaldson': 1, 'Ringo Lam': 1, 'Richard Boddington': 1}), 'Dylan Bak
er': defaultdict(<class 'int'>, {'Tarsem Singh': 1, 'Clare Kilner': 1, 'Bob Odenk
irk': 1}), 'Scott Glenn': defaultdict(<class 'int'>, {'Randall Wallace': 1, 'John
McTiernan': 1, 'Lawrence Kasdan': 1, 'David Hackl': 1}), 'Charlyne Yi': defaultdi
ct(<class 'int'>, {'Judd Apatow': 1}), 'Craig Parker': defaultdict(<class 'int'>,
{'Patrick Tatopoulos': 1}), "Heather O'Rourke": defaultdict(<class 'int'>, {'Tobe
Hooper': 1}), 'Danneel Ackles': defaultdict(<class 'int'>, {'Alan Poul': 1, 'Jon
Hurwitz': 1}), 'Ashley Williams': defaultdict(<class 'int'>, {'Luke Greenfield':
1}), "Jodi Lyn O'Keefe": defaultdict(<class 'int'>, {'': 1}), "Tatum O'Neal": def
aultdict(<class 'int'>, {'Michael Ritchie': 1}), 'Emilia Fox': defaultdict(<class
'int'>, {'Roman Polanski': 1}), 'Keisha Castle-Hughes': defaultdict(<class 'in
t'>, {'Catherine Hardwicke': 1, 'Niki Caro': 1}), "Robert Ri'chard": defaultdict
(<class 'int'>, {'Jaume Collet-Serra': 1}), 'Aaliyah': defaultdict(<class 'int'>,
{'Michael Rymer': 1}), 'Brian Blessed': defaultdict(<class 'int'>, {'Mike Hodge
s': 1, 'Kenneth Branagh': 1}), 'Cicely Tyson': defaultdict(<class 'int'>, {'Rob C
ohen': 1, 'Wayne Wang': 1, 'Bryan Barber': 1, 'Darren Grant': 1}), 'Daniel Mays':
defaultdict(<class 'int'>, {'Susanna White': 1}), 'Frank Whaley': defaultdict(<cl
ass 'int'>, {'Danny DeVito': 1}), 'Mitch Pileggi': defaultdict(<class 'int'>, {'C
hris Carter': 1}), 'Ken Arnold': defaultdict(<class 'int'>, {'John Singleton':
1}), 'Drake Bell': defaultdict(<class 'int'>, {'Craig Mazin': 1, 'Stephen Frear
s': 1, 'Taedong Park': 1, 'Deb Hagan': 1}), 'Alona Tal': defaultdict(<class 'in
t'>, {'Allen Hughes': 1}), 'Shannon Elizabeth': defaultdict(<class 'int'>, {'Wes
Craven': 1, 'Steve Beck': 1, 'Gregory Poirier': 1}), 'Jim Gaffigan': defaultdict
(<class 'int'>, {'Anne Fletcher': 1, 'George Ratliff': 1, 'Sol Tryon': 1}), 'Alex
Russell': defaultdict(<class 'int'>, {'Kimble Rendall': 1}), 'Rosanna Arquette':
defaultdict(<class 'int'>, {'John Milius': 1}), 'Tim Meadows': defaultdict(<class
'int'>, {'Jake Kasdan': 1}), 'Gregory Smith': defaultdict(<class 'int'>, {'Les Ma
yfield': 1, 'Bronwen Hughes': 1, 'Jason Eisener': 1}), 'Michiel Huisman': default
dict(<class 'int'>, {'Jean-Marc Vallée': 2, 'Paul Verhoeven': 1}), 'Paul Sorvin
o': defaultdict(<class 'int'>, {'Hark Tsui': 1, 'Warren Beatty': 1, 'Émile Gaudre
ault': 1, 'Darren Lynn Bousman': 1}), 'Mireille Enos': defaultdict(<class 'int'>,
{'David Ayer': 1}), 'Kevin Zegers': defaultdict(<class 'int'>, {'Peter Hewitt':
1, 'Rob Schmidt': 1, 'Charles Martin Smith': 1, 'Derick Martini': 1, 'Duncan Tuck
er': 1}), 'Marley Shelton': defaultdict(<class 'int'>, {'Ronny Yu': 1, 'Jamie Bla
nks': 1, 'David Caffrey': 1}), 'Archie Panjabi': defaultdict(<class 'int'>, {'Rid

```

```

ley Scott': 1, 'Fernando Meirelles': 1, 'Gurinder Chadha': 1, "Damien O'Donnell":
1)), 'Ruth Wilson': defaultdict(<class 'int'>, {'': 1}), 'Oliver Reed': defaultdi
ct(<class 'int'>, {'Moustapha Akkad': 1, 'Carol Reed': 1, 'Guy Hamilton': 1}), 'J
eremy W. Auman': defaultdict(<class 'int'>, {'Ang Lee': 1}), 'Maricel Álvarez': d
efaultdict(<class 'int'>, {'Alejandro G. Iñárritu': 1}), 'Richard E. Grant': defa
ultdict(<class 'int'>, {'Martin Scorsese': 1, 'Uli Edel': 1, 'Richard Shepard':
1}), 'David Paymer': defaultdict(<class 'int'>, {'Hugh Johnson': 1, 'Rob Reiner':
1}), 'Rumi Hiiragi': defaultdict(<class 'int'>, {'Hayao Miyazaki': 1}), 'Tiago Ri
ani': defaultdict(<class 'int'>, {'George Tillman Jr.': 1}), 'Liam Aiken': default
tdict(<class 'int'>, {'Hugh Hudson': 1, 'John Hoffman': 1, 'Nicholas Hytner': 1,
'Michael Winterbottom': 1}), 'Terry Pheto': defaultdict(<class 'int'>, {'Justin C
hadwick': 1, 'Bille August': 1, 'Gavin Hood': 1}), 'Nia Long': defaultdict(<class
'int'>, {'Raja Gosnell': 1, 'Rupert Wainwright': 1, 'Brian Levant': 1, 'Mennan Ya
po': 1, 'Peter Atencio': 1, 'F. Gary Gray': 1}), 'Kyle Hebert': defaultdict(<clas
s 'int'>, {'Jeremy Degruzon': 1}), 'Ryan Guzman': defaultdict(<class 'int'>, {'Sc
ott Speer': 1, 'Rob Cohen': 1, 'Rita Merson': 1}), 'Lin Shaye': defaultdict(<clas
s 'int'>, {'David R. Ellis': 1, 'Leigh Whannell': 1, 'James Wan': 2, 'Stiles Whit
e': 1, 'William Eubank': 1}), 'Laura Harring': defaultdict(<class 'int'>, {'Jonat
han Hensleigh': 1, 'Glen Morgan': 1}), 'Ryan Kruger': defaultdict(<class 'int'>,
{'Neil Marshall': 1}), 'James Coburn': defaultdict(<class 'int'>, {'Brian Levan
t': 1, 'Sam Peckinpah': 2, 'John Sturges': 1}), 'Ana de la Reguera': defaultdict
(<class 'int'>, {'Jared Hess': 1}), 'Matt Lucas': defaultdict(<class 'int'>, {'Pa
ul Feig': 1}), 'Brandon T. Jackson': defaultdict(<class 'int'>, {'John Whitesel
l': 1, 'Erik White': 1}), 'Illeana Douglas': defaultdict(<class 'int'>, {'Frank M
arshall': 1, 'David Koepp': 1}), 'Callum Rennie': defaultdict(<class 'int'>, {'Je
an-Pierre Jeunet': 1, 'Les Mayfield': 1, 'Christopher Nolan': 1}), 'Ken Howard':
defaultdict(<class 'int'>, {'John Gatins': 1, 'Irwin Winkler': 1}), 'Frances Fish
er': defaultdict(<class 'int'>, {'Peter Howitt': 1, 'Vadim Perelman': 1, 'Catheri
ne Hardwicke': 1, 'Karen Moncrieff': 1}), 'Alice Braga': defaultdict(<class 'in
t'>, {'Miguel Sapochnik': 1, 'David Mamet': 1, 'Fernando Meirelles': 1}), 'George
Kennedy': defaultdict(<class 'int'>, {'Mark Dindal': 1, 'Peter Segal': 1, 'Rupert
Wyatt': 1, 'David Zucker': 1, 'Sam Levinson': 1, 'Michael Gornick': 1}), 'Joan Pl
owright': defaultdict(<class 'int'>, {'Tom Reeve': 1}), 'Rob Riggle': defaultdict
(<class 'int'>, {'Nanette Burstein': 1, 'Luke Greenfield': 1, 'Neal Brennan':
1}), 'Alicia Witt': defaultdict(<class 'int'>, {'Stephen Herek': 1, 'Tyler Perr
y': 1, 'Jamie Blanks': 1, 'Mike Binder': 1, 'John Waters': 1}), 'Desmond Llewely
n': defaultdict(<class 'int'>, {'Lewis Gilbert': 1, 'Terence Young': 1, 'Guy Hami
lton': 1}), 'Sharlto Copley': defaultdict(<class 'int'>, {'Neill Blomkamp': 1}),
'John Doman': defaultdict(<class 'int'>, {'Clint Eastwood': 1}), 'Emily Osment':
defaultdict(<class 'int'>, {'Peter Chelsom': 1}), 'Elizabeth Daily': defaultdict
(<class 'int'>, {'Stig Bergqvist': 1, 'Igor Kovalyov': 1, 'John Eng': 1, '': 1}),
'George Carlin': defaultdict(<class 'int'>, {'Barbra Streisand': 1}), 'Hope Davi
s': defaultdict(<class 'int'>, {'Alexander Payne': 1, 'Lasse Hallström': 1}), 'Co
ry Hardrict': defaultdict(<class 'int'>, {'Jonathan Levine': 1}), 'Zach Gilford':
defaultdict(<class 'int'>, {'': 2, 'Jee-woon Kim': 1, 'Matt Bettinelli-Olpin':
1}), 'Josh Charles': defaultdict(<class 'int'>, {'John Singleton': 1, 'Tim Hill':
1}), 'Tina Fey': defaultdict(<class 'int'>, {'Michael McCullers': 1, 'Shawn Lev
y': 1, 'Mark Waters': 1, 'Paul Weitz': 1, 'Matt Maiellaro': 1}), 'Antony Starr':
defaultdict(<class 'int'>, {'Steven Brill': 1}), 'Kurt Fuller': defaultdict(<clas
s 'int'>, {'Woody Allen': 1}), 'Brenda Song': defaultdict(<class 'int'>, {'John S
chultz': 1}), 'Patrick Macnee': defaultdict(<class 'int'>, {'John Glen': 1, 'Joe
Dante': 1}), 'Mark Gatiss': defaultdict(<class 'int'>, {'Steve Box': 1, 'Jez Butt
erworth': 1}), 'Kate Upton': defaultdict(<class 'int'>, {'Bobby Farrelly': 1}),
'Mehcad Brooks': defaultdict(<class 'int'>, {'James Gartner': 1, 'H.M. Coakley':
1}), 'Edgar Arreola': defaultdict(<class 'int'>, {'Denis Villeneuve': 1}), 'Alyso
n Stoner': defaultdict(<class 'int'>, {'Jon M. Chu': 1}), 'Faizon Love': defaulttd
ict(<class 'int'>, {'John Stockwell': 1, 'Martin Lawrence': 1}), 'Danny Huston':
defaultdict(<class 'int'>, {'David Slade': 1}), 'Carmen Electra': defaultdict(<cl
ass 'int'>, {'Jason Friedberg': 2, 'David Zucker': 1}), 'Yaphet Kotto': defaulttdi
ct(<class 'int'>, {'Paul Michael Glaser': 1, 'Guy Hamilton': 1, 'Michael Roemer':

```

```

1)), 'Jessica Barden': defaultdict(<class 'int'>, {'Joe Wright': 1}), 'Brian Thom
pson': defaultdict(<class 'int'>, {'John R. Leonetti': 1}), 'Debi Mazar': default
dict(<class 'int'>, {'': 1}), 'Devin Ratray': defaultdict(<class 'int'>, {'Martha
Coolidge': 1, 'Alexander Payne': 1, 'Jeremy Saulnier': 1}), 'Craig Ferguson': def
aultdict(<class 'int'>, {'Stephen J. Anderson': 1, 'Kevin Allen': 1}), 'Warren Be
atty': defaultdict(<class 'int'>, {'Warren Beatty': 1}), 'Tika Sumpter': defaultd
ict(<class 'int'>, {'Tate Taylor': 1}), 'Rainn Wilson': defaultdict(<class 'in
t'>, {'Ivan Reitman': 1}), 'Sarah Polley': defaultdict(<class 'int'>, {'Vincenzo
Natali': 1, 'Doug Liman': 1, 'Isabel Coixet': 1}), 'Eddie Cibrian': defaultdict(<
class 'int'>, {'Bruce Hunt': 1, 'J.B. Rogers': 1, 'Tyler Perry': 1, 'Bill Duke':
1}), 'Jessica Paré': defaultdict(<class 'int'>, {'Paul McGuigan': 1}), 'Grace Phi
pps': defaultdict(<class 'int'>, {'Craig Gillespie': 1}), 'Saffron Burrows': defa
ultdict(<class 'int'>, {'Chris Roberts': 1}), 'Ian Whyte': defaultdict(<class 'in
t'>, {'James Wong': 1}), 'Annabella Sciorra': defaultdict(<class 'int'>, {'Andy C
adiff': 1}), 'Ólafur Darri Ólafsson': defaultdict(<class 'int'>, {'': 1}), 'Tyler
Labine': defaultdict(<class 'int'>, {'Peter Howitt': 1}), 'Emir Kusturica': defau
ltdict(<class 'int'>, {'Neil Jordan': 1, 'Patrice Leconte': 1}), 'Emily Watson':
defaultdict(<class 'int'>, {'Chris Noonan': 1, 'Alan Parker': 1, 'Brian Perciva
l': 1, 'Richard E. Grant': 1}), 'Dong-gun Jang': defaultdict(<class 'int'>, {'Kai
ge Chen': 1}), 'Steve Howey': defaultdict(<class 'int'>, {'Corey Yuen': 1, 'Bo Ze
nga': 1}), 'Louise Bourgoin': defaultdict(<class 'int'>, {'Laurent Tirard': 1}),
'Paz Vega': defaultdict(<class 'int'>, {'Olivier Dahan': 1}), 'Konstantin Khabens
kiy': defaultdict(<class 'int'>, {'Maksim Fadeev': 1, 'Timur Bekmambetov': 1, 'Al
eksandr Veledinskiy': 1}), 'Sandra Ellis Lafferty': defaultdict(<class 'int'>,
{'James Mangold': 1, 'Ryan Smith': 1}), 'Amanda Schull': defaultdict(<class 'in
t'>, {'': 1, 'Nicholas Hytner': 1}), 'Lisa Edelstein': defaultdict(<class 'int'>,
{'Edward Norton': 1}), 'Dick Clark': defaultdict(<class 'int'>, {'George Cloone
y': 1}), 'Aldis Hodge': defaultdict(<class 'int'>, {'F. Gary Gray': 1}), 'Joe Lo
Truglio': defaultdict(<class 'int'>, {'David Wain': 1, 'Steve Pink': 1, 'Frank Se
bastiano': 1, 'Matt Walsh': 1}), 'Eddie Cahill': defaultdict(<class 'int'>, {"Gav
in O'Connor": 1}), 'Shun Oguri': defaultdict(<class 'int'>, {'Shinji Aramaki':
1}), 'Julian Glover': defaultdict(<class 'int'>, {'John Glen': 1, 'Dominic Burn
s': 1}), 'Lilli Lavigne': defaultdict(<class 'int'>, {'Peter Chelsom': 1}), 'Mia S
ara': defaultdict(<class 'int'>, {'Peter Hyams': 1}), 'David Lyons': defaultdict
(<class 'int'>, {'Lasse Hallström': 1}), 'Vinessa Shaw': defaultdict(<class 'in
t'>, {'Kenny Ortega': 1}), 'Zoë Kravitz': defaultdict(<class 'int'>, {'Scott Hick
s': 1, 'Anna Boden': 1, 'Analeine Cal y Mayor': 1}), 'Elizabeth McGovern': defaul
tdict(<class 'int'>, {'Matthew Vaughn': 1}), 'Talulah Riley': defaultdict(<class
'int'>, {'Joe Wright': 1}), 'Philip Baker Hall': defaultdict(<class 'int'>, {'Jan
usz Kaminski': 1, 'John Dahl': 1}), 'Jacques Perrin': defaultdict(<class 'int'>,
{'Jacques Perrin': 1}), 'Samaire Armstrong': defaultdict(<class 'int'>, {'Donald
Petrie': 1}), 'Scott Grimes': defaultdict(<class 'int'>, {'Jay Roach': 1}), 'Mara
Wilson': defaultdict(<class 'int'>, {'Michael Ritchie': 1, 'Britt Allcroft': 1}),
'Dominic Flores': defaultdict(<class 'int'>, {'David Gordon Green': 1}), 'Matt Sm
ith': defaultdict(<class 'int'>, {'Burr Steers': 1}), 'Tenzin Thuthob Tsarong': d
efaultdict(<class 'int'>, {'Martin Scorsese': 1}), 'Louis Jourdan': defaultdict(<
class 'int'>, {'John Glen': 1}), 'Harold Ramis': defaultdict(<class 'int'>, {'Jud
d Apatow': 1}), 'Richard Wilson': defaultdict(<class 'int'>, {'David Lean': 1}),
'Phil Davis': defaultdict(<class 'int'>, {'Richard Eyre': 1}), 'Ellen Burstyn': d
efaultdict(<class 'int'>, {'Callie Khouri': 1, 'William Friedkin': 1, 'Tim Chambe
rs': 1, 'Darren Aronofsky': 1}), 'John Belushi': defaultdict(<class 'int'>, {'Joh
n Landis': 2}), 'America Ferrera': defaultdict(<class 'int'>, {'Sanaa Hamri': 1,
'Rick Famuyiwa': 1, 'Patricia Cardoso': 1, 'Patricia Rikken': 1}), 'Dolly Parto
n': defaultdict(<class 'int'>, {'Todd Graff': 1, 'Colin Higgins': 1}), 'Sharon Sm
all': defaultdict(<class 'int'>, {'Chris Weitz': 1}), 'Gerald McRaney': defaulttdi
ct(<class 'int'>, {'Wolfgang Petersen': 1}), 'Jackie Gleason': defaultdict(<class
'int'>, {'': 1}), 'Piper Laurie': defaultdict(<class 'int'>, {'Walter Murch':
1}), 'Harry Lennix': defaultdict(<class 'int'>, {'E. Elias Merhige': 1, 'Kevin Ro
dney Sullivan': 1, 'Rod Lurie': 1}), 'Meat Loaf': defaultdict(<class 'int'>, {'Ro
nny Yu': 1, 'Uwe Boll': 1, 'Antonio Banderas': 1}), 'Kim-Marie Woodhouse': defaul

```

```

tddict(<class 'int'>, {'Kenneth Branagh': 1}), 'Erick Avari': defaultdict(<class
'int'>, {'Steve Carr': 1}), 'Rob McElhenney': defaultdict(<class 'int'>, {'':
1}), 'Ryan O'Neal': defaultdict(<class 'int'>, {'Richard Attenborough': 1, 'Stanl
ey Kubrick': 1}), 'Robert Pine': defaultdict(<class 'int'>, {'Wes Craven': 1}),
'Sarah Carter': defaultdict(<class 'int'>, {'David R. Ellis': 1}), 'Daniel Roebuc
k': defaultdict(<class 'int'>, {'Kevin Allen': 1, 'Harald Zwart': 1, 'George Gall
o': 1, 'James Wong': 1, 'Don Coscarelli': 1}), 'Pam Ferris': defaultdict(<class
'int'>, {'James McTeigue': 1}), 'Rip Torn': defaultdict(<class 'int'>, {'Donald P
etrie': 1, 'Tom Green': 1, 'Ira Sachs': 1}), 'Christopher Lambert': defaultdict(<
class 'int'>, {'Andrew Morahan': 1, 'Paul W.S. Anderson': 1, 'Russell Mulcahy':
1, 'Douglas Aarniokoski': 1, 'Stuart Gordon': 1, 'Timothy Hines': 1, 'Luc Besso
n': 1}), 'David Conrad': defaultdict(<class 'int'>, {'Michael Cohn': 1}), 'Woody
Allen': defaultdict(<class 'int'>, {'Woody Allen': 10}), 'Paul Hogan': defaultdic
t(<class 'int'>, {'Alan Shapiro': 1, 'John Cornell': 1, 'Peter Faiman': 1}), 'Chr
isto Jivkov': defaultdict(<class 'int'>, {'Mel Gibson': 1}), 'Luke Bracey': defau
ltdict(<class 'int'>, {'Michael Hoffman': 1, 'Roger Donaldson': 1}), 'Katharine I
sabelle': defaultdict(<class 'int'>, {'Ronny Yu': 1, 'Cassandra Nicolaou': 1}),
'Cara Delevingne': defaultdict(<class 'int'>, {'Michael Winterbottom': 1}), 'Mike
Vogel': defaultdict(<class 'int'>, {'Matt Reeves': 1, 'Ken Kwapis': 1, 'Jim Field
Smith': 1}), 'Kevin Nash': defaultdict(<class 'int'>, {'Michael Pressman': 1}),
'Bruce McGill': defaultdict(<class 'int'>, {'Tim Story': 1, 'Kevin Brodie': 1, 'T
im Boxell': 1}), 'Lena Olin': defaultdict(<class 'int'>, {'Lasse Hallström': 1,
'Jaume Balagueró': 1}), 'Mary Lynn Rajs kub': defaultdict(<class 'int'>, {'Charles
Herman-Wurmfeld': 1, '': 1, 'Colin Trevorrow': 1}), 'Bailee Madison': defaultdict
(<class 'int'>, {'Andy Fickman': 1, 'Troy Nixey': 1}), 'Lee Remick': defaultdict
(<class 'int'>, {'Richard Donner': 1}), 'Chelan Simmons': defaultdict(<class 'in
t'>, {'James Wong': 1}), 'Blythe Danner': defaultdict(<class 'int'>, {'Scott Hick
s': 1, 'Peter H. Hunt': 1}), 'Darcy Donavan': defaultdict(<class 'int'>, {'Adam M
cKay': 1}), 'Heather Locklear': defaultdict(<class 'int'>, {'Brett Ratner': 1, 'M
ark L. Lester': 1, 'Jim Amatulli': 1}), 'Jim Meskimen': defaultdict(<class 'in
t'>, {'Paul Thomas Anderson': 1}), 'Dave Chappelle': defaultdict(<class 'int'>,
{'Malcolm D. Lee': 1, 'Tamra Davis': 1}), 'Nicki Aycox': defaultdict(<class 'in
t'>, {'Victor Salva': 1}), 'Lily Atkinson': defaultdict(<class 'int'>, {'Steve Be
ndelack': 1}), 'Nicholas Gonzalez': defaultdict(<class 'int'>, {'Dwight H. Littl
e': 1, 'Jeff Crook': 1, 'Richard Montoya': 1}), 'Fanny Ardant': defaultdict(<clas
s 'int'>, {'Shekhar Kapur': 1}), 'Alexis Arquette': defaultdict(<class 'int'>,
{'Ronny Yu': 1}), 'Joseph Cotten': defaultdict(<class 'int'>, {'Richard Fleische
r': 1, 'King Vidor': 1, 'Henry Hathaway': 1}), 'Ari Graynor': defaultdict(<class
'int'>, {'David Gordon Green': 1, 'Peter Sollett': 1, 'Lee Toland Krieger': 1}),
'Jere Burns': defaultdict(<class 'int'>, {'Simon Wincer': 1}), 'Connie Ray': defa
ultdict(<class 'int'>, {'Tim Fywell': 1}), 'Aimee Garcia': defaultdict(<class 'in
t'>, {'Nigel Cole': 1, 'Emily Dell': 1}), 'Justin Fletcher': defaultdict(<class
'int'>, {'': 1}), 'Claudia Christian': defaultdict(<class 'int'>, {'Don Michael P
aul': 1}), 'Tyler James Williams': defaultdict(<class 'int'>, {'Paul Feig': 1, 'T
ina Gordon Chism': 1}), 'Dianne Wiest': defaultdict(<class 'int'>, {'James Bridge
s': 1, 'Herbert Ross': 1, 'Jessie Nelson': 1, 'Lawrence Kasdan': 1, 'Joel Schumac
her': 1, 'John Cameron Mitchell': 1}), 'Stephen Dillane': defaultdict(<class 'in
t'>, {'Bill Paxton': 1}), 'Laura Ramsey': defaultdict(<class 'int'>, {'Carter Smi
th': 1, 'Renny Harlin': 1, 'Rick de Oliveira': 1}), 'Mark Valley': defaultdict(<c
lass 'int'>, {'John Schlesinger': 1}), 'Frank Grillo': defaultdict(<class 'int'>,
{'Wes Craven': 1, 'James DeMonaco': 1, 'Will Canon': 1}), 'Chris Marquette': defa
ultdict(<class 'int'>, {'Luke Greenfield': 1}), 'Martin Starr': defaultdict(<clas
s 'int'>, {'Bruce McCulloch': 1}), 'Sissy Spacek': defaultdict(<class 'int'>, {'A
kiva Schaffer': 1, 'Michael Apted': 1, 'Courtney Solomon': 1, 'David Lynch': 1}),
'Chris Zylka': defaultdict(<class 'int'>, {'David R. Ellis': 1}), 'Frank Finlay':
defaultdict(<class 'int'>, {'Tobe Hooper': 1, 'Martin Ritt': 1}), 'Marco Treviñ
o': defaultdict(<class 'int'>, {'Patricia Rikken': 1}), 'Victor Wong': defaultdic
t(<class 'int'>, {'John Carpenter': 1, 'Charles T. Kanganis': 1, 'Jim Goddard':
1, 'Michael Ritchie': 1}), 'Dany Boon': defaultdict(<class 'int'>, {'Pascal Chaum
eil': 1, 'Dany Boon': 1}), 'Tom Berenger': defaultdict(<class 'int'>, {'Ron Maxwe

```



```

ll': 1, 'Lance Hool': 1}}, 'John Doe': defaultdict(<class 'int'>, {'Jonathan Kapl
an': 1}}, 'Kyra Sedgwick': defaultdict(<class 'int'>, {'Ole Bornedal': 1, 'Sidney
Lumet': 1}}, 'Alexander Skarsgård': defaultdict(<class 'int'>, {'Rod Lurie': 1,
'Lars von Trier': 1, 'Zal Batmanglij': 1, 'Marielle Heller': 1}}, 'Brian Boswort
h': defaultdict(<class 'int'>, {'Craig R. Baxley': 1}}, 'Tanya Roberts': defaulttdi
ct(<class 'int'>, {'John Guillermin': 1}}, 'Cheech Marin': defaultdict(<class 'i
nt'>, {'Marcos Siega': 1, 'José Luis Valenzuela': 1}}, 'Donald Faison': defaulttdi
ct(<class 'int'>, {'Jeffrey W. Byrd': 1, 'Amy Heckerling': 1, 'Bille Woodruff':
1, 'Scott Foley': 1, 'David Raynr': 1}}, 'Don McKellar': defaultdict(<class 'in
t'>, {'Fernando Meirelles': 1}}, 'Dallas Barnett': defaultdict(<class 'int'>, {'A
ndrew Adamson': 1}}, 'Zoe Kazan': defaultdict(<class 'int'>, {'Richard Linklate
r': 1, 'Bradley Rust Gray': 1}}, 'François Cluzet': defaultdict(<class 'int'>,
{'Guillaume Canet': 1, 'Thomas Lilti': 1}}, 'Scout Taylor-Compton': defaultdict(<
class 'int'>, {'Taylor Hackford': 1, 'Rob Zombie': 1}}, 'Yolande Moreau': default
dict(<class 'int'>, {'Jérôme Deschamps': 1}}, 'Sergey Garmash': defaultdict(<clas
s 'int'>, {'Inna Evlannikova': 1}}, 'Rod Steiger': defaultdict(<class 'int'>, {'S
ergey Bondarchuk': 1}}, 'Siu-Wong Fan': defaultdict(<class 'int'>, {'Teddy Chan':
1}}, 'Minnie Driver': defaultdict(<class 'int'>, {'Bonnie Hunt': 1, 'Trey Parke
r': 1, 'Hayao Miyazaki': 1, 'George Armitage': 1, 'Oliver Parker': 1, 'Mel Smit
h': 1, 'Gina Prince-Bythewood': 1}}, 'Gerry Bednob': defaultdict(<class 'int'>,
{'Kevin Smith': 1}}, 'Lena Horne': defaultdict(<class 'int'>, {'Sidney Lumet':
1}}, 'Willie Garson': defaultdict(<class 'int'>, {'Ron Shelton': 1, 'Spike Jonz
e': 1}}, 'Donald Pleasence': defaultdict(<class 'int'>, {'Richard Williams': 1,
'Lewis Gilbert': 1, 'John Carpenter': 1, 'Dominique Othenin-Girard': 1, 'Joe Chap
pelle': 1, 'Dwight H. Little': 1}}, 'Seymour Cassel': defaultdict(<class 'int'>,
{'Michael Dinner': 1, 'Randal Kleiser': 1}}, 'Judith Barsi': defaultdict(<class
'int'>, {'Joseph Sargent': 1, 'Don Bluth': 1}}, 'Tasha Smith': defaultdict(<class
'int'>, {'Fred Durst': 1}}, 'Alessandro Nivola': defaultdict(<class 'int'>, {'Ann
e Fontaine': 1}}, 'Tom Burke': defaultdict(<class 'int'>, {'Stephen Frears': 1}},
'Cole Hauser': defaultdict(<class 'int'>, {'': 1, 'Paul Abascal': 1}}, 'Romola Ga
rai': defaultdict(<class 'int'>, {'Mira Nair': 1, 'Christian Volckman': 1, 'Ruair
i Robinson': 1, '': 1}}, 'Bingbing Fan': defaultdict(<class 'int'>, {'Teddy Cha
n': 1, 'Tung-Shing Yee': 1}}, 'Drew Powell': defaultdict(<class 'int'>, {'Mikael
Håfström': 1}}, 'Joan Rivers': defaultdict(<class 'int'>, {'Mel Brooks': 1}}, 'Ce
m Yilmaz': defaultdict(<class 'int'>, {'Russell Crowe': 1}}, 'Manish Dayal': defa
ultdict(<class 'int'>, {'Lasse Hallström': 1}}, 'Roma Downey': defaultdict(<class
'int'>, {'Christopher Spencer': 1}}, "Denis O'Hare": defaultdict(<class 'int'>,
{'Mikael Håfström': 1}}, 'Carlos Ponce': defaultdict(<class 'int'>, {'Mike Bigelo
w': 1, 'Linda Mendoza': 1}}, 'Jack Reynor': defaultdict(<class 'int'>, {'Ken Scot
t': 1}}, 'Radivoje Bukvic': defaultdict(<class 'int'>, {'Susanna White': 1}}, 'Am
anda Detmer': defaultdict(<class 'int'>, {'Dennis Dugan': 1}}, 'Zachary Gordon':
defaultdict(<class 'int'>, {'David Bowers': 2, 'Rian Johnson': 1}}, 'Rock Hudso
n': defaultdict(<class 'int'>, {'Blake Edwards': 1}}, 'Ed Skrein': defaultdict(<c
lass 'int'>, {'Camille Delamarre': 1}}, 'Gary Lewis': defaultdict(<class 'int'>,
{'Christian Carion': 1}}, 'Romain Duris': defaultdict(<class 'int'>, {'Laurent Ti
rard': 1, 'Cédric Klapisch': 1}}, 'Pablo Rago': defaultdict(<class 'int'>, {'Juan
José Campanella': 1}}, 'Ornella Muti': defaultdict(<class 'int'>, {'Woody Allen':
1}}, 'Paige Turco': defaultdict(<class 'int'>, {'Stuart Gillard': 1}}, 'Hunter Pa
rrish': defaultdict(<class 'int'>, {'Richard LaGravenese': 1, 'David Jacobson':
1}}, 'Omari Hardwick': defaultdict(<class 'int'>, {'Tyler Perry': 1, 'Salim Aki
l': 1, 'Benny Boom': 1, 'Ava DuVernay': 1}}, 'Jason London': defaultdict(<class
'int'>, {'Katt Shea': 1}}, 'Chris Harrison': defaultdict(<class 'int'>, {'': 1}},
'Elaine Hendrix': defaultdict(<class 'int'>, {'John Duigan': 1, 'Jim Issa': 1}},
'Bob Balaban': defaultdict(<class 'int'>, {'Steven Spielberg': 1}}, 'Ian Somerhal
der': defaultdict(<class 'int'>, {'Jim Sonzero': 1, 'Irwin Winkler': 1, 'Roger Av
ary': 1}}, 'Dean Stockwell': defaultdict(<class 'int'>, {'Tony Scott': 1, 'Gregor
Jordan': 1, 'Elia Kazan': 1}}, 'Angus T. Jones': defaultdict(<class 'int'>, {'Ada
m Shankman': 1}}, 'Bam Margera': defaultdict(<class 'int'>, {'Jeff Tremaine':
3}}, 'Mackenzie Foy': defaultdict(<class 'int'>, {'James Wan': 1, 'Stéphane Aubie
r': 1}}, 'Miranda Cosgrove': defaultdict(<class 'int'>, {'Richard Linklater': 1,

```

```

'Scott Marshall': 1}}, 'Shailene Woodley': defaultdict(<class 'int'>, {'Alexander
Payne': 1, 'Josh Boone': 1, 'James Ponsoldt': 1}}, 'Leonard Roberts': defaultdict
(<class 'int'>, {'Charles Stone III': 1, 'Theodore Witcher': 1}}, 'Sam Claflin':
defaultdict(<class 'int'>, {'Thea Sharrock': 1}}, 'John Amos': defaultdict(<class
'int'>, {'Tyler Perry': 1, 'Dario Argento': 1, 'Melvin Van Peebles': 1}}, 'Barry
Watson': defaultdict(<class 'int'>, {'Stephen Kay': 1}}, 'Kath Soucie': defaulttdi
ct(<class 'int'>, {'Jun Falkenstein': 1, 'Frank Nissen': 1}}, 'Lamman Rucker': de
faultdict(<class 'int'>, {'': 1}}, 'Billy Crudup': defaultdict(<class 'int'>, {'T
om McCarthy': 1, 'Bart Freundlich': 1, 'William H. Macy': 1, 'Alison Maclean':
1}}, 'Richard Lawson': defaultdict(<class 'int'>, {'Kevin Rodney Sullivan': 1}},
'Violante Placido': defaultdict(<class 'int'>, {'Anton Corbijn': 1}}, 'Jon Seda':
defaultdict(<class 'int'>, {'Gregory Nava': 1}}, 'Diedrich Bader': defaultdict(<c
lass 'int'>, {'Jason Friedberg': 1, 'Chuck Sheetz': 1}}, 'Jeremy Shada': defaultd
ict(<class 'int'>, {'Trey Parker': 1}}, 'Gunnar Hansen': defaultdict(<class 'in
t'>, {'John Luessenhop': 1, 'Tobe Hooper': 1}}, 'William McNamara': defaultdict(<
class 'int'>, {'Jon Amiel': 1, 'Tanner Beard': 1, 'Jay Alaimo': 1, 'David DeCotea
u': 1}}, 'Marisa Saks': defaultdict(<class 'int'>, {'Malcolm D. Lee': 1}}, 'Peter
Firth': defaultdict(<class 'int'>, {'Kevin Reynolds': 1}}, 'Isaiah Washington': d
efaultdict(<class 'int'>, {'Steve Beck': 1}}, 'Sela Ward': defaultdict(<class 'in
t'>, {'Nelson McCormick': 1}}, 'Johnny Lewis': defaultdict(<class 'int'>, {'Eric
Valette': 1}}, 'Kay Panabaker': defaultdict(<class 'int'>, {'Andrew Fleming':
1}}, 'Jay Laga'aia': defaultdict(<class 'int'>, {'Michael Spierig': 1}}, 'Leighto
n Meester': defaultdict(<class 'int'>, {'Thomas Bezucha': 1, 'Christian E. Christ
iansen': 1, 'Shana Feste': 1}}, 'Kevin Sussman': defaultdict(<class 'int'>, {'Ant
hony Bell': 1}}, 'Kurtwood Smith': defaultdict(<class 'int'>, {'Gus Van Sant': 1,
'Miguel Arteta': 1, 'Steve James': 1}}, 'Samira Wiley': defaultdict(<class 'in
t'>, {'Henry Joost': 1}}, 'Sheryl Lee': defaultdict(<class 'int'>, {'John Carpent
er': 1}}, 'Pablo Sevilla': defaultdict(<class 'int'>, {'': 1}}, 'Janet Leigh': de
faultdict(<class 'int'>, {'Alfred Hitchcock': 1, 'Jack Smight': 1}}, 'Mako': defa
ultdict(<class 'int'>, {'Steve Rash': 1, 'Richard Fleischer': 1}}, 'Shengyi Huan
g': defaultdict(<class 'int'>, {'Stephen Chow': 1}}, 'Leah Remini': defaultdict(<
class 'int'>, {'': 1}}, 'Mustafa Haidari': defaultdict(<class 'int'>, {'Marc Fors
ter': 1}}, 'Thomas Lennon': defaultdict(<class 'int'>, {'Chris Koch': 1}}, 'Andy
Dick': defaultdict(<class 'int'>, {'Amy Heckerling': 1}}, 'Ronald Reagan': defaul
tdict(<class 'int'>, {'Michael Moore': 1}}, 'Andrew Fiscella': defaultdict(<class
'int'>, {'Nimród Antal': 1, 'John Erick Dowdle': 1}}, 'James Callis': defaultdict
(<class 'int'>, {'Michael O. Sajbel': 1}}, 'Aasif Mandvi': defaultdict(<class 'in
t'>, {'David Koepp': 1}}, 'Taylor Cole': defaultdict(<class 'int'>, {'Renny Harli
n': 1}}, 'Quinton Aaron': defaultdict(<class 'int'>, {'Michel Gondry': 1}}, 'Aish
a Tyler': defaultdict(<class 'int'>, {'James Wan': 1}}, 'Scott Baio': defaultdict
(<class 'int'>, {'Bob Clark': 1}}, 'Debbie Reynolds': defaultdict(<class 'int'>,
{'Michael Lembeck': 1}}, 'Sammy Davis Jr.': defaultdict(<class 'int'>, {'Bob Foss
e': 1}}, 'Martin Dew': defaultdict(<class 'int'>, {'Paul Thomas Anderson': 1}},
'Chris Brown': defaultdict(<class 'int'>, {'Benson Lee': 1, 'Sylvain White': 1,
'Tim Story': 1}}, 'Chriss Anglin': defaultdict(<class 'int'>, {'David Zucker':
1}}, 'Kevin Alejandro': defaultdict(<class 'int'>, {'Fred Wolf': 1}}, 'Catherine
Lough Haggquist': defaultdict(<class 'int'>, {'Uwe Boll': 1}}, 'Cameron Bright':
defaultdict(<class 'int'>, {'Jonathan Glazer': 1}}, 'Landon Liboiron': defaultdic
t(<class 'int'>, {'Paul Gross': 1}}, 'Dylan Authors': defaultdict(<class 'int'>,
{'Marc Abraham': 1}}, 'Isabelle Adjani': defaultdict(<class 'int'>, {'Jean-Paul R
appeneau': 1}}, 'Steve Guttenberg': defaultdict(<class 'int'>, {'Nancy Walker':
1, 'Barry Levinson': 1, 'Hugh Wilson': 1}}, 'Donny Boaz': defaultdict(<class 'in
t'>, {'Angelo Pizzo': 1}}, 'Damon Wayans': defaultdict(<class 'int'>, {'Richard B
enjamin': 1}}, 'William Hootkins': defaultdict(<class 'int'>, {'Katsuhiko Ôtomo':
1}}, 'Esai Morales': defaultdict(<class 'int'>, {'Kevin Reynolds': 1, 'Luis Valde
z': 1, 'Kristin Rizzo': 1}}, 'Jiao Xu': defaultdict(<class 'int'>, {'Stephen Cho
w': 1}}, 'Christian Clavier': defaultdict(<class 'int'>, {'Jean-Marie Poiré':
2}}, 'Marco St. John': defaultdict(<class 'int'>, {'Kevin Munroe': 1}}, 'Peter Va
ughan': defaultdict(<class 'int'>, {'': 1}}, 'Necar Zadegan': defaultdict(<class
'int'>, {'Jane Clark': 1}}, 'Andy Lau': defaultdict(<class 'int'>, {'Daniel Lee':

```

```

1, 'Alan Yuen': 1, 'Chia-Liang Liu': 1}}, 'Mani Bharathi': defaultdict(<class 'int'>, {'Gnana Rajasekaran': 1}), 'Joey D. Vieira': defaultdict(<class 'int'>, {'Tom Walsh': 1}), 'Moritz Bleibtreu': defaultdict(<class 'int'>, {'Uli Edel': 1, 'Tom Tykwer': 1}), 'Mary McDonnell': defaultdict(<class 'int'>, {'Kevin Costner': 1}), 'Dan Byrd': defaultdict(<class 'int'>, {'Mark Rosman': 1, 'Alexandre Aja': 1}), 'David Arquette': defaultdict(<class 'int'>, {'Wes Craven': 1, 'Finn Taylor': 1}), 'Scatman Crothers': defaultdict(<class 'int'>, {'Stanley Kubrick': 1, 'Milos Forman': 1}), 'Jeffrey Combs': defaultdict(<class 'int'>, {'William Malone': 1}), 'Eric Mendenhall': defaultdict(<class 'int'>, {'Tyler Perry': 1}), 'Kari J. Kramer': defaultdict(<class 'int'>, {'Rob Hawk': 1}), 'Taran Killam': defaultdict(<class 'int'>, {'Shawn Levy': 1, 'Jerome Elston Scott': 1}), 'Mem Ferda': defaultdict(<class 'int'>, {'Lee Tamahori': 1, 'Julian Gilbey': 1}), "Nicole 'Snooki' Polizzi": defaultdict(<class 'int'>, {'': 1}), 'Bunta Sugawara': defaultdict(<class 'int'>, {'Hayao Miyazaki': 1}), 'Katrina Bowden': defaultdict(<class 'int'>, {'Sean Anders': 1, 'Eli Craig': 1, 'Malcolm Goodwin': 1}), 'Darci Kistler': defaultdict(<class 'int'>, {'Emile Ardolino': 1}), 'Tony Nappo': defaultdict(<class 'int'>, {'George A. Romero': 1}), 'Louis Koo': defaultdict(<class 'int'>, {'Danny Pang': 1}), 'Craig Roberts': defaultdict(<class 'int'>, {'Nicholas Stoller': 1, 'Richard Ayoade': 1}), 'Luenell': defaultdict(<class 'int'>, {'Larry Charles': 1}), 'Linda Cardellini': defaultdict(<class 'int'>, {'Robert Luketic': 1, 'Nicholaus Gossens': 1, 'James Gunn': 1}), 'Craig T. Nelson': defaultdict(<class 'int'>, {'Thomas Bezucha': 1, 'Jeff Kanew': 1, '': 1}), 'Nia Vardalos': defaultdict(<class 'int'>, {'Kirk Jones': 1, 'Donald Petrie': 1, 'Joel Zwick': 1}), 'Shane Black': defaultdict(<class 'int'>, {'John McTiernan': 1}), 'Jeffrey Jones': defaultdict(<class 'int'>, {'Milos Forman': 1, 'Don Michael Paul': 1}), 'Raoul Bova': defaultdict(<class 'int'>, {'Audrey Wells': 1}), 'Kristin Scott Thomas': defaultdict(<class 'int'>, {'Robert Altman': 1, 'Gilles Paquet-Brenner': 1, 'Mike Newell': 1, 'Sam Taylor-Johnson': 1}), 'Peter Gallagher': defaultdict(<class 'int'>, {'': 1, 'Max Mayer': 1, 'Steven Soderbergh': 1}), 'Sally Hawkins': defaultdict(<class 'int'>, {'Woody Allen': 1, 'Phil Morrison': 1}), 'Christian Clemenson': defaultdict(<class 'int'>, {'Paul Greengrass': 1}), 'Christian Monzon': defaultdict(<class 'int'>, {'Bille Woodruff': 1}), 'Hulk Hogan': defaultdict(<class 'int'>, {'Rick Friedberg': 1}), 'Bob Newhart': defaultdict(<class 'int'>, {'Mike Nichols': 1}), 'Collette Wolfe': defaultdict(<class 'int'>, {'Jody Hill': 1}), 'Nicholas Rowe': defaultdict(<class 'int'>, {'Barry Levinson': 1}), 'Kelsey Grammer': defaultdict(<class 'int'>, {'Kevin Tancharoen': 1}), 'Alan Ruck': defaultdict(<class 'int'>, {'Chris Columbus': 1, 'Daniel Hsia': 1}), 'Jason Robards': defaultdict(<class 'int'>, {'William A. Fraker': 1}), 'Nancy Allen': defaultdict(<class 'int'>, {'Brian De Palma': 1}), 'Rebecca Budig': defaultdict(<class 'int'>, {'Courtney Solomon': 1}), 'Joan Allen': defaultdict(<class 'int'>, {'Ang Lee': 1, 'Lenny Abrahamson': 1}), 'Yaya DaCosta': defaultdict(<class 'int'>, {'Rob Reiner': 1}), 'David Margulies': defaultdict(<class 'int'>, {'Adrian Lyne': 1}), 'Ian McDiarmid': defaultdict(<class 'int'>, {'Matthew Robbins': 1}), 'David Masterson': defaultdict(<class 'int'>, {'Michael Hoffman': 1}), 'Tadanobu Asano': defaultdict(<class 'int'>, {'Sergey Bodrov': 1}), 'Barry Bostwick': defaultdict(<class 'int'>, {'Hal Needham': 1}), 'Julie Christie': defaultdict(<class 'int'>, {'Kenneth Branagh': 1, 'David Lean': 1}), 'Poppy Montgomery': defaultdict(<class 'int'>, {'': 1}), 'Ulrich Tukur': defaultdict(<class 'int'>, {'Michael Haneke': 1}), 'James Hetfield': defaultdict(<class 'int'>, {'Nimród Antal': 1}), 'Austin Nichols': defaultdict(<class 'int'>, {'Gregor Jordan': 1}), 'Edgar Ramírez': defaultdict(<class 'int'>, {'': 1}), 'Takuya Kimura': defaultdict(<class 'int'>, {'Tran Anh Hung': 1, 'Takashi Yamazaki': 1}), 'Greg Nicotero': defaultdict(<class 'int'>, {'George A. Romero': 1}), 'Ernie Reyes Jr.': defaultdict(<class 'int'>, {'Richard Fleischer': 1}), 'Vanessa Ferlito': defaultdict(<class 'int'>, {'Tyler Perry': 1}), 'Jason Momoa': defaultdict(<class 'int'>, {'David Hayter': 1}), 'Cassie Ventura': defaultdict(<class 'int'>, {'Jon M. Chu': 1}), 'Henry Ian Cusick': defaultdict(<class 'int'>, {'Xavier Gens': 1, 'Philip Saville': 1, 'Stephen Herek': 1}), 'Jennifer Hudson': defaultdict(<class 'int'>, {'Kasi Lemmons': 1, 'Darrell Roodt': 1}), 'Kirk Fox': defaultdict(<class 'int'>, {'Matt Dillon': 1}), 'Fionnula Flanagan': defaultdict(<class 'int'>, {'Alejandro Amenábar': 1}), 'Jeremy Brett': defaultdict(<class 'int'>, {'Geo

```

```

rge Cukor': 1}}, 'Chris Moss': defaultdict(<class 'int'>, {'Jay Chandrasekhar':
1}}, 'Emmanuelle Vaugier': defaultdict(<class 'int'>, {'Michael Lehmann': 1, 'Dar
ren Lynn Bousman': 1}}, 'Kevin Greivoux': defaultdict(<class 'int'>, {'Sam Weisma
n': 1}}, 'Morgan Saylor': defaultdict(<class 'int'>, {'Niki Caro': 1}}, 'T.I.': d
efaultdict(<class 'int'>, {'Chris Robinson': 1}}, 'Thomas F. Wilson': defaultdict
(<class 'int'>, {'Trent Cooper': 1, 'Fred Walton': 1}}, 'Ava Gardner': defaultdic
t(<class 'int'>, {'Nicholas Ray': 1}}, 'Lou Taylor Pucci': defaultdict(<class 'in
t'>, {'Fede Alvarez': 1}}, 'DeRay Davis': defaultdict(<class 'int'>, {'Ian Iqbal
Rashid': 1}}, 'Aidan Quinn': defaultdict(<class 'int'>, {'Rowdy Herrington': 1,
'Rick Bieber': 1, 'Conor McPherson': 1, 'Maggie Greenwald': 1, 'Mary Pat Kelly':
1}}, 'Tippi Hedren': defaultdict(<class 'int'>, {'Noel Marshall': 1}}, 'Brenda Fr
icker': defaultdict(<class 'int'>, {'Joel Schumacher': 1}}, 'Julianna Guill': def
aultdict(<class 'int'>, {'Todd Lincoln': 1}}, 'Hal Holbrook': defaultdict(<class
'int'>, {'Oliver Stone': 1}}, 'Danny Strong': defaultdict(<class 'int'>, {'Joe Nu
ssbaum': 1}}, 'Bruno Kirby': defaultdict(<class 'int'>, {'Rob Reiner': 1}}, 'Etha
n Suplee': defaultdict(<class 'int'>, {'Todd Phillips': 1, 'Tony Kaye': 1, 'Kevin
Smith': 2, 'Morgan J. Freeman': 1}}, 'B  r  nice Bejo': defaultdict(<class 'int'>,
{'Michel Hazanavicius': 1}}, 'Clive Russell': defaultdict(<class 'int'>, {'Cyrus
Nowrasteh': 1}}, 'Jennifer Esposito': defaultdict(<class 'int'>, {'Perry Andelin
Blake': 1, 'Daniel Taplitz': 1}}, 'Ivana Baquero': defaultdict(<class 'int'>, {'G
uillermo del Toro': 1}}, 'Leslie Bibb': defaultdict(<class 'int'>, {'John Whitese
ll': 1, 'Sam Miller': 1, 'Goran Dukic': 1, 'Catherine Jelski': 1}}, 'Brittany Dan
iel': defaultdict(<class 'int'>, {'Dennie Gordon': 1, 'Jay Chandrasekhar': 1}},
'Bolo Yeung': defaultdict(<class 'int'>, {'Sheldon Lettich': 1, 'Newt Arnold':
1}}, 'Bill Bailey': defaultdict(<class 'int'>, {'Edgar Wright': 1}}, 'Gustaf Skar
sg  rd': defaultdict(<class 'int'>, {'Peter Flinth': 1}}, 'Gretchen Mol': defaultd
ict(<class 'int'>, {'Josef Rusnak': 1, 'Jason Alexander': 1}}, 'Farrah Fawcett':
defaultdict(<class 'int'>, {'Lance Rivera': 1, 'Michael Anderson': 1}}, 'Lochlyn
Munro': defaultdict(<class 'int'>, {'Bruce Paltrow': 1, 'Fred Savage': 1}}, 'Alis
on Lohman': defaultdict(<class 'int'>, {'Susanne Bier': 1, 'Michael Mayer': 1}},
'Charles Napier': defaultdict(<class 'int'>, {'Kenneth Johnson': 1, 'Russ Meyer':
1}}, 'Paz de la Huerta': defaultdict(<class 'int'>, {'Gaspar No  ': 1}}, 'Steven V
an Zandt': defaultdict(<class 'int'>, {'': 1}}, 'David Alpay': defaultdict(<class
'int'>, {'Atom Egoyan': 1}}, 'Bruce Dern': defaultdict(<class 'int'>, {'William B
indley': 1, 'Joe Dante': 1, 'Douglas Trumbull': 1}}, 'Carole Bouquet': defaultdic
t(<class 'int'>, {'G  rard Krawczyk': 1}}, 'Dustin Milligan': defaultdict(<class
'int'>, {'James Gunn': 1}}, 'Chen Chang': defaultdict(<class 'int'>, {'Ang Lee':
1}}, 'Bex Taylor-Klaus': defaultdict(<class 'int'>, {'': 1}}, 'Nicole Dionne': de
faultdict(<class 'int'>, {'JK Youn': 1}}, "Beverly D'Angelo": defaultdict(<class
'int'>, {'Harold Ramis': 1, 'John Turturro': 1}}, 'Jackson Nicoll': defaultdict(<
class 'int'>, {'Jeff Tremaine': 1}}, 'Roger Allam': defaultdict(<class 'int'>,
{'Stephen Frears': 1}}, 'Frankie Muniz': defaultdict(<class 'int'>, {'Shawn Lev
y': 1}}, 'Clayton Snyder': defaultdict(<class 'int'>, {'Jim Fall': 1}}, 'Jordana
Brewster': defaultdict(<class 'int'>, {'Robert Rodriguez': 1, 'Angela Robinson':
1}}, "Rae'Ven Kelly": defaultdict(<class 'int'>, {'Brian Gibson': 1}}, 'Claude Ra
ins': defaultdict(<class 'int'>, {'David Lean': 1, 'Frank Capra': 1}}, 'Tony Gold
wyn': defaultdict(<class 'int'>, {'Dennis Iliadis': 1, 'Tom McLoughlin': 1}}, 'Ja
ck Davenport': defaultdict(<class 'int'>, {'Clare Kilner': 1}}, 'Jessie T. Ushe
r': defaultdict(<class 'int'>, {'Thomas Carter': 1}}, 'Kim Director': defaultdict
(<class 'int'>, {'Joe Berlinger': 1}}, 'Christine Taylor': defaultdict(<class 'in
t'>, {'Andrew Fleming': 1}}, 'Sierra McCormick': defaultdict(<class 'int'>, {'Eli
zabeth Allen Rosenbaum': 1}}, 'Kieran Culkin': defaultdict(<class 'int'>, {'Rober
t Harmon': 1, 'Peter Care': 1, 'Burr Steers': 1, 'Peter Chelsom': 1}}, 'Jeff Kobe
r': defaultdict(<class 'int'>, {'Martin Weisz': 1}}, 'William Hurt': defaultdict
(<class 'int'>, {'Jay Russell': 1, 'L  a Pool': 1, 'Stephen Hopkins': 1, 'Randall
Miller': 1}}, 'Kelly Carlson': defaultdict(<class 'int'>, {'John Bonito': 1}}, 'J
ason Priestley': defaultdict(<class 'int'>, {'Stephan Elliott': 1, 'Richard Kwiet
niowski': 1}}, 'Yuri Lowenthal': defaultdict(<class 'int'>, {'Mike Nawrocki':
1}}, 'Daniel Stern': defaultdict(<class 'int'>, {'Drew Barrymore': 1, 'Douglas Ch
eek': 1}}, 'Trevor Morgan': defaultdict(<class 'int'>, {'Steve Gomer': 1}}, 'Oliv

```

```

er James': defaultdict(<class 'int'>, {'Sean McNamara': 1}), 'David Ogden Stier
s': defaultdict(<class 'int'>, {'': 2}), 'Khandi Alexander': defaultdict(<class
'int'>, {'Ron Shelton': 1, 'Leon Ichaso': 1}), 'Elizabeth Berrington': defaultdic
t(<class 'int'>, {'Martin McDonagh': 1}), 'Larry David': defaultdict(<class 'in
t'>, {'Woody Allen': 1, 'Joshua Tickell': 1}), 'Matthew Settle': defaultdict(<cla
ss 'int'>, {'Mary Lambert': 1}), 'Levon Helm': defaultdict(<class 'int'>, {'Tommy
Lee Jones': 1}), 'Anna Silk': defaultdict(<class 'int'>, {'': 1}), 'Thomas Kretsc
hmann': defaultdict(<class 'int'>, {'Brad Anderson': 1, 'Oliver Hirschbiegel':
1}), 'Curtis Armstrong': defaultdict(<class 'int'>, {'Michael Chapman': 1, 'Doug
Atchison': 1, 'Stephen Sommers': 1}), 'Thomas Stoppel': defaultdict(<class 'in
t'>, {'Khalil Sullins': 1}), 'Brigitte Helm': defaultdict(<class 'int'>, {'Fritz
Lang': 1}), 'David Belle': defaultdict(<class 'int'>, {'Pierre Morel': 1}), 'Qi S
hu': defaultdict(<class 'int'>, {'Hsiao-Hsien Hou': 1}), 'Konstantin Lavronenko':
defaultdict(<class 'int'>, {'Andrey Zvyagintsev': 1}), 'Nirut Sirichanya': defaul
tdict(<class 'int'>, {'Tony Jaa': 1}), 'Christopher Heyerdahl': defaultdict(<clas
s 'int'>, {'Russell Mulcahy': 1}), 'Eddie Izzard': defaultdict(<class 'int'>, {'S
tefan Ruzowitzky': 1, 'E. Elias Merhige': 1}), 'Stephen Graham': defaultdict(<cla
ss 'int'>, {'Jake Paltrow': 1, 'Roland Suso Richter': 1, 'Tom Hooper': 1, 'Shane
Meadows': 1}), 'Anna Friel': defaultdict(<class 'int'>, {'Juraj Jakubisko': 1, 'D
avid Leland': 1, 'Barry Levinson': 1}), 'Birgitte Hjort Sørensen': defaultdict(<c
lass 'int'>, {'Gabe Ibáñez': 1}), 'Bruce Payne': defaultdict(<class 'int'>, {'Ger
ry Lively': 1}), 'Maximilian Dirr': defaultdict(<class 'int'>, {'Daniele Luchett
i': 1}), 'Samantha Morton': defaultdict(<class 'int'>, {'Julien Temple': 1, 'Mich
ael Winterbottom': 1, 'Lynne Ramsay': 1, 'Liv Ullmann': 1, 'Eric Styles': 1}), 'S
haron Leal': defaultdict(<class 'int'>, {'Peter Cousins': 1}), 'Anil Kapoor': def
aultdict(<class 'int'>, {'Danny Boyle': 1}), 'Fred Gwynne': defaultdict(<class 'i
nt'>, {'Adrian Lyne': 1, 'Jonathan Lynn': 1}), 'María Valverde': defaultdict(<cla
ss 'int'>, {'Vidhu Vinod Chopra': 1}), 'Doug E. Doug': defaultdict(<class 'int'>,
{'Jon Turteltaub': 1}), 'Jensen Ackles': defaultdict(<class 'int'>, {'Patrick Lus
sier': 1, '': 1}), 'Caroline Munro': defaultdict(<class 'int'>, {'Lewis Gilbert':
1}), 'Amy Bruni': defaultdict(<class 'int'>, {'': 1}), 'Phil Vischer': defaultdic
t(<class 'int'>, {'Mike Nawrocki': 1}), 'Janet Jackson': defaultdict(<class 'in
t'>, {'John Singleton': 1}), 'Mike Epps': defaultdict(<class 'int'>, {'Kevin Bra
y': 1, 'Marcus Raboy': 1}), 'Linda Blair': defaultdict(<class 'int'>, {'John Boor
man': 1}), 'Lily Tomlin': defaultdict(<class 'int'>, {'Franco Zeffirelli': 1, 'Da
vid O. Russell': 1}), 'Peter Mensah': defaultdict(<class 'int'>, {'James Isaac':
1, '': 1}), 'Monica Potter': defaultdict(<class 'int'>, {'Mark Waters': 1}), 'Tho
mas McDonnell': defaultdict(<class 'int'>, {'Josh Schwartz': 1}), 'Emmanuelle Seig
ner': defaultdict(<class 'int'>, {'Julian Schnabel': 1}), 'Derek Luke': defaulttdi
ct(<class 'int'>, {'Phillip Noyce': 1}), 'Melissa Sagemiller': defaultdict(<class
'int'>, {'Stephen Carpenter': 1}), 'Seth Gilliam': defaultdict(<class 'int'>, {'J
ames Ivory': 1}), 'Lesley Manville': defaultdict(<class 'int'>, {'Mike Leigh':
2}), 'Mathieu Amalric': defaultdict(<class 'int'>, {'Alain Resnais': 1, 'Mathieu
Amalric': 1}), 'Sebastian Koch': defaultdict(<class 'int'>, {'Costa-Gavras': 1,
'Florian Henckel von Donnersmarck': 1}), 'Woo-sung Jung': defaultdict(<class 'in
t'>, {'Chao-Bin Su': 1}), 'John Heard': defaultdict(<class 'int'>, {'Neil Burge
r': 1, 'Ed Harris': 1}), 'Krystyna Janda': defaultdict(<class 'int'>, {'': 1}),
'Jacek Koman': defaultdict(<class 'int'>, {'': 1}), 'Belén Rueda': defaultdict(<c
lass 'int'>, {'Alejandro Amenábar': 1}), 'Jason Mewes': defaultdict(<class 'in
t'>, {'Hyung-rae Shim': 1, 'Kevin Smith': 1}), 'Usher Raymond': defaultdict(<clas
s 'int'>, {'Jon M. Chu': 1}), 'Sean Patrick Thomas': defaultdict(<class 'int'>,
{'Thomas Carter': 1, 'Tim Story': 1}), 'Tuesday Knight': defaultdict(<class 'in
t'>, {'Renny Harlin': 1}), 'Jack Palance': defaultdict(<class 'int'>, {'Christoph
er Cain': 1}), 'Óscar Jaenada': defaultdict(<class 'int'>, {'Jaume Collet-Serra':
1}), 'Chris Elliott': defaultdict(<class 'int'>, {'Chris Koch': 1}), 'Kathleen Tu
rner': defaultdict(<class 'int'>, {'Bob Clark': 1, 'John Waters': 1}), 'Jenny Agu
tter': defaultdict(<class 'int'>, {'John Lafia': 1}), 'Elina Alminas': defaultdic
t(<class 'int'>, {'Alex Garland': 1}), 'Teo Halm': defaultdict(<class 'int'>, {'D
ave Green': 1}), 'Yuki Matsuzaki': defaultdict(<class 'int'>, {'Clint Eastwood':
1}), 'Don Rickles': defaultdict(<class 'int'>, {'Bob Saget': 1}), 'Michael Nyqvist

```

```

t': defaultdict(<class 'int'>, {'Niels Arden Oplev': 1, 'Kay Pollak': 1}), 'Justin Chon': defaultdict(<class 'int'>, {'Jon Lucas': 1, 'Alex Craig Mann': 1}), 'Burt Young': defaultdict(<class 'int'>, {'Jon Hess': 1}), 'John Boyega': defaultdict(<class 'int'>, {'Joe Cornish': 1}), 'Jelena Jovanova': defaultdict(<class 'int'>, {'Angelina Jolie Pitt': 1}), 'Steve Irwin': defaultdict(<class 'int'>, {'John Stainton': 1}), 'Dameon Clarke': defaultdict(<class 'int'>, {'Glenn Ficarra': 1}), 'Rae Dawn Chong': defaultdict(<class 'int'>, {'Jean-Jacques Annaud': 1}), 'Gabriel Millman': defaultdict(<class 'int'>, {'Michael Hoffman': 1}), 'Julian Morris': defaultdict(<class 'int'>, {'Stewart Hendler': 1, 'Darin Scott': 1, 'Jeff Wadlow': 1, 'Oliver Blackburn': 1}), 'David Niven': defaultdict(<class 'int'>, {'John Cromwell': 1}), 'Richard Johnson': defaultdict(<class 'int'>, {'Mark Herman': 1}), 'Isabella Rossellini': defaultdict(<class 'int'>, {'Abel Ferrara': 1, 'James Gray': 1, 'Paul Fierlinger': 1}), 'Min-sik Choi': defaultdict(<class 'int'>, {'Jeon-kyu Kang': 1, 'Chan-wook Park': 2}), 'Sean Young': defaultdict(<class 'int'>, {'Tom Shadyac': 1}), 'Alex Veadov': defaultdict(<class 'int'>, {'Mike McCoy': 1}), 'Shelley Duvall': defaultdict(<class 'int'>, {'Terry Gilliam': 1}), 'Britney Spears': defaultdict(<class 'int'>, {'Tamra Davis': 1}), 'Dax Flame': defaultdict(<class 'int'>, {'Nima Nourizadeh': 1}), 'George C. Scott': defaultdict(<class 'int'>, {'Franklin J. Schaffner': 1, 'Robert Rossen': 1, 'Stanley Kubrick': 1}), 'Steven Mackintosh': defaultdict(<class 'int'>, {'Brian Henson': 1}), 'Lexie Contursi': defaultdict(<class 'int'>, {'Dan Cutforth': 1}), 'Talia Shire': defaultdict(<class 'int'>, {'John Frankenheimer': 1}), 'Tammin Sursok': defaultdict(<class 'int'>, {'Elizabeth Allen Rosenbaum': 1}), 'Nat Wolff': defaultdict(<class 'int'>, {'Jake Schreier': 1}), 'Common': defaultdict(<class 'int'>, {'Sanaa Hamri': 1, 'Michel Gondry': 1}), 'Jürgen Prochnow': defaultdict(<class 'int'>, {'Wolfgang Petersen': 1}), 'Heather Matarazzo': defaultdict(<class 'int'>, {'Wallace Wolodarsky': 1, 'Todd Solondz': 1}), 'Tom Hughes': defaultdict(<class 'int'>, {'Richard Curtis': 1}), 'Nate Parker': defaultdict(<class 'int'>, {'Nicholas Jarecki': 1, 'Gina Prince-Bythewood': 1}), 'Gary Weeks': defaultdict(<class 'int'>, {'Dean Israelite': 1}), 'Melanie Griffith': defaultdict(<class 'int'>, {'Woody Allen': 1, 'Brian De Palma': 1}), 'Jill Clayburgh': defaultdict(<class 'int'>, {'Ryan Murphy': 1}), 'Anika Noni Rose': defaultdict(<class 'int'>, {'Robert Iscove': 1}), 'Donna Bae': defaultdict(<class 'int'>, {'Joon-ho Bong': 1}), 'Jeremy Davies': defaultdict(<class 'int'>, {'Antonia Bird': 1, 'Steven Shainberg': 1}), 'Toni Servillo': defaultdict(<class 'int'>, {'Paolo Sorrentino': 1}), 'Mia Wasikowska': defaultdict(<class 'int'>, {'Chan-wook Park': 1, 'Rodrigo García': 1}), 'David Richmond-Peck': defaultdict(<class 'int'>, {'Ira Sachs': 1}), 'Eamonn Walker': defaultdict(<class 'int'>, {'Carroll Ballard': 1}), 'Eva Amurri Martino': defaultdict(<class 'int'>, {'Vadim Perelman': 1}), 'Marcell Nagy': defaultdict(<class 'int'>, {'Lajos Koltai': 1}), 'Honglei Sun': defaultdict(<class 'int'>, {'Yimou Zhang': 1}), 'David Hewlett': defaultdict(<class 'int'>, {'Vincenzo Natali': 3}), 'Ye Liu': defaultdict(<class 'int'>, {'Chuan Lu': 1}), 'Richard Coyle': defaultdict(<class 'int'>, {'Renny Harlin': 1}), 'Rachael Carpani': defaultdict(<class 'int'>, {'Christopher Smith': 1}), 'Jacob Vargas': defaultdict(<class 'int'>, {'Randall Wallace': 1}), 'Gary Daniels': defaultdict(<class 'int'>, {'David Winters': 1}), 'Miko Hughes': defaultdict(<class 'int'>, {'Mary Lambert': 1, 'Wes Craven': 1}), 'Tatsuo Matsumura': defaultdict(<class 'int'>, {'Akira Kurosawa': 1}), 'Wei Tang': defaultdict(<class 'int'>, {'Mabel Cheung': 1}), 'Phoebe Cates': defaultdict(<class 'int'>, {'Joe Dante': 1}), 'Wagner Moura': defaultdict(<class 'int'>, {'Stephen Daldry': 1, 'José Padilha': 1, 'Karim Aïnouz': 1}), 'Lucas Grabeel': defaultdict(<class 'int'>, {'Kenny Ortega': 3}), 'Zoe Ventoura': defaultdict(<class 'int'>, {'Brian Trenchard-Smith': 1}), 'Corbin Bernsen': defaultdict(<class 'int'>, {'David S. Ward': 1, 'Rusty Cundieff': 1, '': 1, 'Corbin Bernsen': 1, 'Daniel Mellitz': 1}), 'Lily Cole': defaultdict(<class 'int'>, {'Oliver Parker': 1}), 'Lauren German': defaultdict(<class 'int'>, {'Adam Shankman': 1}), 'Lois Smith': defaultdict(<class 'int'>, {'Tim Robbins': 1}), 'Madeline Kahn': defaultdict(<class 'int'>, {'Mel Brooks': 4, 'James Frawley': 1}), 'Steven Bauer': defaultdict(<class 'int'>, {'Brian De Palma': 1}), 'Louise Fletcher': defaultdict(<class 'int'>, {'Tobe Hooper': 1}), 'Lee Majors': defaultdict(<class 'int'>, {'Brendan Malloy': 1}), 'Luke Newberry': defaultdict(<class 'int'>, {'Dustin Hoffman': 1}), 'Patrici

```

```

a Tallman': defaultdict(<class 'int'>, {'Sam Raimi': 1}), 'Michael McKean': default
ldict(<class 'int'>, {'Daisy von Scherler Mayer': 1, 'Christopher Guest': 1}),
'Phyllida Law': defaultdict(<class 'int'>, {'Agnieszka Holland': 1}), 'Sam Rile
y': defaultdict(<class 'int'>, {'Rowan Joffe': 1, 'Anton Corbijn': 1}), 'John Cas
savetes': defaultdict(<class 'int'>, {'Giuliano Montaldo': 1, 'Roman Polanski':
1}), 'Nora Dunn': defaultdict(<class 'int'>, {'Lisa Azuelos': 1}), 'John Howard':
defaultdict(<class 'int'>, {'Ray Lawrence': 1}), 'Shah Rukh Khan': defaultdict(<c
lass 'int'>, {'Karan Johar': 2, 'Yash Chopra': 2}), 'Keir Dullea': defaultdict(<c
lass 'int'>, {'Stanley Kubrick': 1}), 'Henry Thomas': defaultdict(<class 'int'>,
{'Steven Spielberg': 1, 'Greg Marcks': 1, 'Michael Landon Jr.': 1}), 'Ricky Schro
der': defaultdict(<class 'int'>, {'Floyd Mutrux': 1, 'Jeb Stuart': 1}), 'Antonia
Thomas': defaultdict(<class 'int'>, {'': 1}), 'Arjun Rampal': defaultdict(<class
'int'>, {'Sajid Khan': 1}), 'Jon Gries': defaultdict(<class 'int'>, {'Christopher
Cain': 1}), 'Santiago Cabrera': defaultdict(<class 'int'>, {'Dean Wright': 1}),
'Karin Viard': defaultdict(<class 'int'>, {'Eric Lartigau': 1}), 'Boris Kodjoe':
defaultdict(<class 'int'>, {'Tyler Perry': 1, 'Douglas Aarniokoski': 1, 'Rick Fam
uyiwa': 1, 'Bille Woodruff': 1}), 'Richard Burton': defaultdict(<class 'int'>,
{'Ken Annakin': 1, 'Henry Koster': 1}), 'Lane Smith': defaultdict(<class 'int'>,
{'Stephen Herek': 1}), 'Tina Turner': defaultdict(<class 'int'>, {'George Mille
r': 1}), 'Lauren Cohan': defaultdict(<class 'int'>, {'William Brent Bell': 1}),
'Jeff Fahey': defaultdict(<class 'int'>, {'Brett Leonard': 1}), 'Laila Haley': de
faultdict(<class 'int'>, {'Ciarán Foy': 1}), 'Billy Drago': defaultdict(<class 'i
nt'>, {'Joseph Zito': 1}), 'Jurnee Smollett-Bell': defaultdict(<class 'int'>, {'M
alcolm D. Lee': 1, 'Kasi Lemmons': 1}), 'David Zayas': defaultdict(<class 'int'>,
{'Colin Strause': 1}), 'Tina Desai': defaultdict(<class 'int'>, {'John Madden':
1}), 'Ben Feldman': defaultdict(<class 'int'>, {'Mark Rosman': 1, 'John Erick Dow
dle': 1}), 'Steve Oedekerk': defaultdict(<class 'int'>, {'Steve Oedekerk': 1}),
'Reba McEntire': defaultdict(<class 'int'>, {'Ron Underwood': 1}), 'Byron Mann':
defaultdict(<class 'int'>, {'James Foley': 1}), 'Wendi McLendon-Covey': defaultdi
ct(<class 'int'>, {'Robert Ben Garant': 1}), 'Harry Styles': defaultdict(<class
'int'>, {'Morgan Spurlock': 1}), 'Chase Ellison': defaultdict(<class 'int'>, {'Ji
m Hanon': 1}), 'Jasper Cole': defaultdict(<class 'int'>, {'Jorma Taccone': 1}),
'Sophie Okonedo': defaultdict(<class 'int'>, {'Stephen Frears': 1}), 'Michaela Co
nlin': defaultdict(<class 'int'>, {'': 1}), 'Daniel Sharman': defaultdict(<class
'int'>, {'Marcus Dunstan': 1}), 'Johannes Silberschneider': defaultdict(<class 'i
nt'>, {'François Girard': 1}), 'Dylan Moran': defaultdict(<class 'int'>, {'David
Schwimmer': 1}), 'Bernadette Peters': defaultdict(<class 'int'>, {'Allan Arkush':
1}), 'Flavio Parenti': defaultdict(<class 'int'>, {'Luca Guadagnino': 1}), 'Rober
t Picardo': defaultdict(<class 'int'>, {'John Putch': 1}), 'Peter Fonda': default
dict(<class 'int'>, {'Steven Soderbergh': 1, 'Victor Nunez': 1}), 'Eric Stoltz':
defaultdict(<class 'int'>, {'Terence Davies': 1, 'Roger Avary': 1}), 'Cliff Rober
tson': defaultdict(<class 'int'>, {'Harley Cokeliss': 1, 'Ralph Nelson': 1}), 'Sc
ott Mechlowicz': defaultdict(<class 'int'>, {'Victor Salva': 1}), 'Gillian Whit
e': defaultdict(<class 'int'>, {'Spike Lee': 1}), 'Eoin Macken': defaultdict(<cla
ss 'int'>, {'Jason Zada': 1}), 'Miranda Richardson': defaultdict(<class 'int'>,
{'David Cronenberg': 1}), 'Naseeruddin Shah': defaultdict(<class 'int'>, {'Rakesh
Roshan': 1, 'Mira Nair': 1}), 'Bashar Rahal': defaultdict(<class 'int'>, {'Joshua
Seftel': 1, 'Uwe Boll': 1}), 'Wei Zhao': defaultdict(<class 'int'>, {'Stephen Cho
w': 1}), 'Vincent Gallo': defaultdict(<class 'int'>, {'Vincent Gallo': 1}), 'Isab
elle Huppert': defaultdict(<class 'int'>, {'Claude Chabrol': 1, 'Michael Haneke':
1}), 'Numan Acar': defaultdict(<class 'int'>, {'Jon Stewart': 1}), 'Kip Pardue':
defaultdict(<class 'int'>, {'Dan Harris': 1}), 'Danny Dyer': defaultdict(<class
'int'>, {'Christopher Smith': 1, 'Justin Kerrigan': 1, 'Dan Reed': 1}), 'Jorge Lu
is Moreno': defaultdict(<class 'int'>, {'Rafa Lara': 1}), 'Adrian Martinez': defa
ultdict(<class 'int'>, {'Julio DePietro': 1}), 'Stephanie Szostak': defaultdict(<
class 'int'>, {'Katherine Dieckmann': 1, 'Paolo Monico': 1}), 'Madison Pettis': d
efaultdict(<class 'int'>, {'William Dear': 1}), 'Nicholas Hamilton': defaultdict
(<class 'int'>, {'Kim Farrant': 1}), 'Drew Fuller': defaultdict(<class 'int'>,
{'Scott Marshall': 1}), 'Jim Carter': defaultdict(<class 'int'>, {'Álex de la Igl
esia': 1}), 'Damian Walshe-Howling': defaultdict(<class 'int'>, {'Andrew Trauck

```

```

i': 1}}, 'Teryl Rothery': defaultdict(<class 'int'>, {'Patrick Lussier': 1}}, 'Shane Pollard': defaultdict(<class 'int'>, {'Robert Adetuyi': 1}}, 'Veronica Ferres': defaultdict(<class 'int'>, {'Paul Schrader': 1}}, 'Ingvar Eggert Sigurðsson': defaultdict(<class 'int'>, {'Benedikt Erlingsson': 1}}, 'Sendhil Ramamurthy': defaultdict(<class 'int'>, {'Gurinder Chadha': 1, '': 1}}, 'Jimi Mistry': defaultdict(<class 'int'>, {'Vic Sarin': 1}}, 'Kang-ho Song': defaultdict(<class 'int'>, {'Jee-woon Kim': 1}}, 'Tyler Mane': defaultdict(<class 'int'>, {'William Phillips': 1}}, 'Danny Pino': defaultdict(<class 'int'>, {'Andy Garcia': 1, '': 1}}, 'Daniella Alonso': defaultdict(<class 'int'>, {'Valeri Milev': 1}}, 'Carmen Maura': defaultdict(<class 'int'>, {'Pedro Almodóvar': 1}}, 'John Vernon': defaultdict(<class 'int'>, {'Gerald Potterton': 1}}, 'Jizelle Jade': defaultdict(<class 'int'>, {'Jared Hess': 1}}, 'Bárbara Mori': defaultdict(<class 'int'>, {'Anurag Basu': 1}}, 'Noah Taylor': defaultdict(<class 'int'>, {'Kriv Stenders': 1, 'Scott Hicks': 1}}, 'Topol': defaultdict(<class 'int'>, {'Norman Jewison': 1}}, 'Riccardo Scamarcio': defaultdict(<class 'int'>, {'Ferzan Ozpetek': 1, 'Maiwenn': 1}}, 'Catherine Hicks': defaultdict(<class 'int'>, {'Tom Holland': 1}}, 'Michael Moore': defaultdict(<class 'int'>, {'Michael Moore': 4}}, 'Noel Gugliemi': defaultdict(<class 'int'>, {'James DeMonaco': 1, 'Youssef Delara': 2, 'Jonas Åkerlund': 1}}, 'Don Ameche': defaultdict(<class 'int'>, {'John Landis': 1}}, 'Kristin Lehman': defaultdict(<class 'int'>, {'Christopher McQuarrie': 1}}, 'Ronn Moss': defaultdict(<class 'int'>, {'': 1}}, 'Jung-min Hwang': defaultdict(<class 'int'>, {'Hong-jin Na': 1}}, 'Saruny Wongkrachang': defaultdict(<class 'int'>, {'Chatrichalerm Yukol': 1}}, 'Lambert Wilson': defaultdict(<class 'int'>, {'Eric Lavaine': 1, 'Xavier Beauvois': 1}}, 'Q'orianka Kilcher': defaultdict(<class 'int'>, {'Marc Forby': 1, 'Hank Braxtan': 1}}, 'Jacqueline McKenzie': defaultdict(<class 'int'>, {'Peter Cattaneo': 1}}, 'Lars Mikkelsen': defaultdict(<class 'int'>, {'Ole Christian Madsen': 1}}, 'Harrison Gilbertson': defaultdict(<class 'int'>, {'Jeremy Sims': 1}}, 'Dee Bradley Baker': defaultdict(<class 'int'>, {'': 1}}, 'David Keith': defaultdict(<class 'int'>, {'Mark Rydell': 1, 'Brian Trenchard-Smith': 1}}, 'Jane Fonda': defaultdict(<class 'int'>, {'Roger Vadim': 1}}, 'Björn Sundquist': defaultdict(<class 'int'>, {'Nils Gaup': 1, 'Tommy Wirkola': 1}}, 'Jacinda Barrett': defaultdict(<class 'int'>, {'Mira Nair': 1}}, 'Mariah Carey': defaultdict(<class 'int'>, {'Vondie Curtis-Hall': 1}}, 'Abigail Spencer': defaultdict(<class 'int'>, {'Tom Elkins': 1}}, 'Yu-mi Jeong': defaultdict(<class 'int'>, {'Woo-Suk Kang': 1}}, 'Robert Redford': defaultdict(<class 'int'>, {'J.C. Chandor': 1}}, 'Mary Elizabeth Mastrantonio': defaultdict(<class 'int'>, {'John Sayles': 1}}, 'Katrina Kaif': defaultdict(<class 'int'>, {'Vipul Amrutlal Shah': 1}}, 'Padraic Delaney': defaultdict(<class 'int'>, {'Ken Loach': 1}}, 'Ranbir Kapoor': defaultdict(<class 'int'>, {'Ayan Mukerji': 1, 'Shimit Amin': 1}}, 'Peter Ustinov': defaultdict(<class 'int'>, {'Mervyn LeRoy': 1}}, 'Necati Sasmaz': defaultdict(<class 'int'>, {'Serdar Akar': 1}}, 'Eleanor Parker': defaultdict(<class 'int'>, {'Robert Wise': 1}}, 'Marshall Bell': defaultdict(<class 'int'>, {'Rob Reiner': 1}}, 'Noah Huntley': defaultdict(<class 'int'>, {'Danny Boyle': 1}}, 'Jennifer Freeman': defaultdict(<class 'int'>, {'Chris Stokes': 1}}, 'James Kyson': defaultdict(<class 'int'>, {'Masayuki Ochiai': 1}}, 'Mackenzie Davis': defaultdict(<class 'int'>, {'Tom Gormican': 1}}, 'Brian Doyle-Murray': defaultdict(<class 'int'>, {'Ken Shapiro': 1}}, 'Tina Gilton': defaultdict(<class 'int'>, {'Justin Tipping': 1}}, 'Telly Savalas': defaultdict(<class 'int'>, {'Peter R. Hunt': 1}}, 'Nick Wechsler': defaultdict(<class 'int'>, {'': 1}}, 'Stephen Collins': defaultdict(<class 'int'>, {'John Schultz': 1}}, 'Alexandra Maria Lara': defaultdict(<class 'int'>, {'James Ivory': 1}}, 'Christopher Nicholas Smith': defaultdict(<class 'int'>, {'Nicole Holofcener': 1}}, 'Cameron Monaghan': defaultdict(<class 'int'>, {'Joe Nussbaum': 1}}, 'Faye Dunaway': defaultdict(<class 'int'>, {'Michael Winner': 1, 'Frank Perry': 1}}, 'Sean Lennon': defaultdict(<class 'int'>, {'Tomm Moore': 1}}, 'Alexia Fast': defaultdict(<class 'int'>, {'Andrew Currie': 1}}, 'Ian Tracey': defaultdict(<class 'int'>, {'Claudia Llosa': 1}}, 'Mitsuo Iwata': defaultdict(<class 'int'>, {'Katsuhiro Ôtomo': 1}}, 'Stephen McHattie': defaultdict(<class 'int'>, {'Charles Binamé': 1, 'Bruce McDonald': 1, 'Boris Rodriguez': 1}}, 'Rachel Ann Mullins': defaultdict(<class 'int'>, {'Keith Parmer': 1}}, 'Leehom Wang': defaultdict(<class 'int'>, {'Dennie Gordon': 1}}, 'Celia Imrie': defaultdict(<class 'int'>, {'Ol Parker': 1}}, 'Mckenna Grac

```



```

e': defaultdict(<class 'int'>, {'Bruce Beresford': 1}), 'Spencer Wilding': default
dict(<class 'int'>, {'James Nunn': 1}), 'Scott Takeda': defaultdict(<class 'in
t'>, {'Michael Winnick': 1}), 'Françoise Yip': defaultdict(<class 'int'>, {'Stanl
ey Tong': 1}), 'Joely Richardson': defaultdict(<class 'int'>, {'Tony Richardson':
1, 'Henry Hobson': 1}), 'Molly Parker': defaultdict(<class 'int'>, {'Paul Gross':
1}), 'Eric Johnson': defaultdict(<class 'int'>, {'Russell Holt': 1}), 'Sandrine K
iberlain': defaultdict(<class 'int'>, {'Claude Miller': 1}), 'Akon': defaultdict
(<class 'int'>, {'Jeta Amata': 1}), 'Saif Ali Khan': defaultdict(<class 'int'>,
{'Siddharth Anand': 1, 'Jugal Hansraj': 1}), 'Max Riemelt': defaultdict(<class 'i
nt'>, {'Dennis Gansel': 1}), 'George Coe': defaultdict(<class 'int'>, {'Robert Ma
rcarelli': 1}), 'Darren Shahnavi': defaultdict(<class 'int'>, {'Ernie Barbarash':
1}), 'Michael Nouri': defaultdict(<class 'int'>, {'Adrian Lyne': 1}), 'Sullivan S
tapleton': defaultdict(<class 'int'>, {'Jonathan Liebesman': 1}), 'MyAnna Burin
g': defaultdict(<class 'int'>, {'Neil Marshall': 1, 'Phil Claydon': 1, 'Ben Wheat
ley': 1}), 'Sid Haig': defaultdict(<class 'int'>, {'Rob Zombie': 3, 'Ward Robert
s': 1}), 'Kimberly Elise': defaultdict(<class 'int'>, {'Rick Famuyiwa': 1}), 'Ber
nard Fox': defaultdict(<class 'int'>, {'Sharron Miller': 1}), 'Michael Stuhlbar
g': defaultdict(<class 'int'>, {'Ethan Coen': 1}), 'Irvin Kershner': defaultdict
(<class 'int'>, {'Martin Scorsese': 1}), 'Jonathan Brandis': defaultdict(<class
'int'>, {'Michael Corrente': 1}), 'Indira Varma': defaultdict(<class 'int'>, {'Gu
rinder Chadha': 1, 'Mira Nair': 1}), 'Roy Billing': defaultdict(<class 'int'>,
{'Phillip Noyce': 1}), 'Michael J. Pollard': defaultdict(<class 'int'>, {'Tony Ma
ylam': 1}), 'Melinda Clarke': defaultdict(<class 'int'>, {'': 1, 'Brian Yuzna':
1}), 'Ezra Miller': defaultdict(<class 'int'>, {'Lynne Ramsay': 1, 'Raymond De Fe
litta': 1}), 'Samantha Barks': defaultdict(<class 'int'>, {'John Stephenson':
1}), 'Nathalie Fay': defaultdict(<class 'int'>, {'Sofia Coppola': 1}), 'Geoff Stu
lts': defaultdict(<class 'int'>, {'Bob Gosse': 1, 'Jennifer Finnigan': 1}), 'Lynn
Cohen': defaultdict(<class 'int'>, {'Shari Springer Berman': 1, 'Frank Whaley':
1}), 'Leonid Yarmolnik': defaultdict(<class 'int'>, {'Aleksey German': 1}), 'Mich
ael Wong': defaultdict(<class 'int'>, {'David Lam': 1}), 'Erin Fitzgerald': defau
ltdict(<class 'int'>, {'Vladlen Barbe': 1}), 'Debi Derryberry': defaultdict(<clas
s 'int'>, {'Richard Rich': 1}), 'Ben Browder': defaultdict(<class 'int'>, {'Rober
t C. Cooper': 1}), 'Gordon Pinsent': defaultdict(<class 'int'>, {'Kim Nguyen':
1}), 'Rex Baker': defaultdict(<class 'int'>, {'Jackie Earle Haley': 1}), 'Heike M
akatsch': defaultdict(<class 'int'>, {'Max Färberböck': 1}), 'Glenn Plummer': def
aultdict(<class 'int'>, {'Damian Nieman': 1}), 'Lubna Azabal': defaultdict(<class
'int'>, {'Denis Villeneuve': 1}), 'Laura Morante': defaultdict(<class 'int'>, {'G
abriele Muccino': 1}), 'Deborah Mailman': defaultdict(<class 'int'>, {'Rachel Per
kins': 1}), 'John Cothran': defaultdict(<class 'int'>, {'John Singleton': 1}), 'R
ita Moreno': defaultdict(<class 'int'>, {'Alan Alda': 1, 'Jerome Robbins': 1, 'St
anley Donen': 1}), 'Angie Dickinson': defaultdict(<class 'int'>, {'Brian De Palm
a': 1}), 'Greg Germann': defaultdict(<class 'int'>, {'Nicole Holofcener': 1}), 'D
avid Wayne': defaultdict(<class 'int'>, {'Robert Wise': 1}), 'Leon': defaultdict
(<class 'int'>, {'Louis Morneau': 1}), 'Merab Ninidze': defaultdict(<class 'in
t'>, {'Caroline Link': 1}), 'Emily Podleski': defaultdict(<class 'int'>, {'Sterli
ng Van Wagenen': 1}), 'Florian Lukas': defaultdict(<class 'int'>, {'Wolfgang Beck
er': 1}), 'Linnea Quigley': defaultdict(<class 'int'>, {'Dan O'Bannon': 1, 'Herb
Freed': 1}), 'Maria Pia Calzone': defaultdict(<class 'int'>, {'': 1}), 'Olivia Ne
wton-John': defaultdict(<class 'int'>, {'Randal Kleiser': 1}), 'Osama bin Laden':
defaultdict(<class 'int'>, {'Michael Moore': 1}), 'Katharine Ross': defaultdict(<
class 'int'>, {'George Roy Hill': 1, 'Michael D. Sellers': 1}), 'Ed Wynn': defaul
tdict(<class 'int'>, {'Robert Stevenson': 1}), 'Lisa Wilcox': defaultdict(<class
'int'>, {'Stephen Hopkins': 1}), 'Tisha Campbell-Martin': defaultdict(<class 'in
t'>, {'Spike Lee': 1}), 'Sammo Kam-Bo Hung': defaultdict(<class 'int'>, {'Sammo K
am-Bo Hung': 1}), 'Hugh M. Hefner': defaultdict(<class 'int'>, {'Zach Cregger':
1, 'Fenton Bailey': 1}), 'Francis Capra': defaultdict(<class 'int'>, {'': 1}), 'J
ohn Cameron Mitchell': defaultdict(<class 'int'>, {'John Cameron Mitchell': 1}),
'Antoni Corone': defaultdict(<class 'int'>, {'Sheldon Lettich': 1}), 'Rachel Ali
g': defaultdict(<class 'int'>, {'Jon Knautz': 1}), 'Chelcie Ross': defaultdict(<c
lass 'int'>, {'David Atkins': 1}), 'William Holden': defaultdict(<class 'int'>,

```

```
{'Sam Peckinpah': 1, 'David Lean': 1}}, 'Lesley-Anne Down': defaultdict(<class 'int'>, {'Michael Crichton': 1}), 'Michael Berryman': defaultdict(<class 'int'>, {'Sylvio Tabet': 1, 'Ruggero Deodato': 1, 'Justin Thomas Ostensen': 1}), 'Benoît Magimel': defaultdict(<class 'int'>, {'Claude Chabrol': 1}), 'James Nesbitt': defaultdict(<class 'int'>, {'Peter Cattaneo': 1, '': 1, 'Kirk Jones': 1, 'Paul Green grass': 1}), 'Noam Chomsky': defaultdict(<class 'int'>, {'Tony Kaye': 1, 'Sut Jha lly': 1}), 'Catherine Bell': defaultdict(<class 'int'>, {'Perry Lang': 1}), 'Pruitt Taylor Vince': defaultdict(<class 'int'>, {'Jake Goldberger': 1}), 'Drew Waters': defaultdict(<class 'int'>, {'William Kaufman': 1}), 'Debbie Harry': defaultdict(<class 'int'>, {'David Cronenberg': 1}), 'Yorick van Wageningen': defaultdict(<class 'int'>, {'Martin Koolhoven': 1}), 'Jeanne Moreau': defaultdict(<class 'int'>, {'John Frankenheimer': 1}), 'Simon Bird': defaultdict(<class 'int'>, {'': 1}), 'Wesley Jonathan': defaultdict(<class 'int'>, {'Preston A. Whitmore II': 1, 'Michael Taliferro': 1}), 'Asin': defaultdict(<class 'int'>, {'Ashish R. Mohan': 1}), 'Jonathan D. Mellor': defaultdict(<class 'int'>, {'Jaume Balagueró': 1}), 'Kayla Jackson': defaultdict(<class 'int'>, {'Stewart Raffill': 1}), 'Alice Krige': defaultdict(<class 'int'>, {'Hugh Hudson': 1, 'Amanda Gusack': 1}), 'Stephen Macht': defaultdict(<class 'int'>, {'William Sachs': 1}), 'Jean-Baptiste Maunier': defaultdict(<class 'int'>, {'Christophe Barratier': 1}), 'Alexis Dziena': defaultdict(<class 'int'>, {'James Cox': 1}), 'Ruby Dee': defaultdict(<class 'int'>, {'Spike Lee': 1}), 'Jean-Pierre Léaud': defaultdict(<class 'int'>, {'Aki Kaurismäki': 1}), 'Philippe Nahon': defaultdict(<class 'int'>, {'Gaspar Noé': 1}), 'Lindsay Kay Hayward': defaultdict(<class 'int'>, {'Hitoshi Matsumoto': 1}), 'Anupam Kher': defaultdict(<class 'int'>, {'Rakeysh Omprakash Mehra': 1}), 'Kim Shaw': defaultdict(<class 'int'>, {'Collin Schiffli': 1}), 'Aksel Hennie': defaultdict(<class 'int'>, {'Morten Tyldum': 1}), 'Russell Tovey': defaultdict(<class 'int'>, {'Jon Wright': 1}), 'Campbell Scott': defaultdict(<class 'int'>, {'Michael McGowan': 1, 'Charles Ferguson': 1}), 'Teresa Wright': defaultdict(<class 'int'>, {'Jeannot Szwarc': 1}), 'Abhishek Bachchan': defaultdict(<class 'int'>, {'Rohan Sippy': 1, 'Arjun Sablok': 1}), 'Johanna Braddy': defaultdict(<class 'int'>, {'Henry Joost': 1}), 'John Saxon': defaultdict(<class 'int'>, {'Chuck Russell': 1}), 'Eugenio Derbez': defaultdict(<class 'int'>, {'Eugenio Derbez': 1}), 'Matt Shively': defaultdict(<class 'int'>, {'Henry Joost': 1}), 'Burt Kwouk': defaultdict(<class 'int'>, {'Blake Edwards': 1}), 'Richard Cabral': defaultdict(<class 'int'>, {'Christopher Landon': 1}), 'James Lafferty': defaultdict(<class 'int'>, {'Mike Flanagan': 1}), 'William Schallert': defaultdict(<class 'int'>, {'George Jackson': 1}), 'Tony Bonner': defaultdict(<class 'int'>, {'George Miller': 1}), 'John Beasley': defaultdict(<class 'int'>, {'Robert Duvall': 1}), 'Gene Tierney': defaultdict(<class 'int'>, {'Michael Curtiz': 1}), 'Kane Hodder': defaultdict(<class 'int'>, {'Rob Hedden': 1, 'Adam Marcus': 1, 'John Carl Buechler': 1, 'Matt Jackson': 1}), 'Maribel Verdú': defaultdict(<class 'int'>, {'Alfonso Cuarón': 1}), 'Peter Serafinowicz': defaultdict(<class 'int'>, {'Edgar Wright': 1}), 'Jessica Lowndes': defaultdict(<class 'int'>, {'Mickey Liddell': 1, 'Shyam Madiraju': 1}), 'Warren Christie': defaultdict(<class 'int'>, {'Gonzalo López-Gallego': 1}), 'Sterling Jerins': defaultdict(<class 'int'>, {'John Erick Dowdle': 1}), 'Vanna Bonta': defaultdict(<class 'int'>, {'Don Coscarelli': 1}), 'Gina Lollobrigida': defaultdict(<class 'int'>, {'King Vidor': 1, 'Carol Reed': 1}), 'Fred Willard': defaultdict(<class 'int'>, {'Michael Tiddes': 1}), 'Gary Cooper': defaultdict(<class 'int'>, {'Frank Borzage': 1, 'Fred Zinnemann': 1}), 'Lara Jill Miller': defaultdict(<class 'int'>, {'Mamoru Hosoda': 1}), 'Marie-Josée Croze': defaultdict(<class 'int'>, {'Denys Arcand': 1}), 'Robert Shaw': defaultdict(<class 'int'>, {'Richard Lester': 1, 'Fred Zinnemann': 1, 'Terence Young': 1}), 'Brendan Fehr': defaultdict(<class 'int'>, {'J.S. Cardone': 1}), 'Fran Drescher': defaultdict(<class 'int'>, {'Jay Levey': 1}), 'Claudia Cardinale': defaultdict(<class 'int'>, {'Sergio Leone': 1}), 'Norbert Ferrer': defaultdict(<class 'int'>, {'Ronan Chapalain': 1}), 'Cassandra Peterson': defaultdict(<class 'int'>, {'Jerry Belson': 1}), 'Kim Dickens': defaultdict(<class 'int'>, {'Jake Kasdan': 1}), 'Dana Delany': defaultdict(<class 'int'>, {'Paul Schrader': 1}), 'Alia Shawkat': defaultdict(<class 'int'>, {'Jeremy Saulnier': 1}), 'Liza Minnelli': defaultdict(<class 'int'>, {'Billy Kent': 1}), 'Andrew Wilson': defaultdict(<class 'int'>, {'Wes Anderson': 1}), 'Akshaye Khanna': defau
```

```

ltdict(<class 'int'>, {'Feroz Abbas Khan': 1}), 'Jeffrey Frost': defaultdict(<class 'int'>, {'Errol Morris': 1}), 'Raymond Burr': defaultdict(<class 'int'>, {'Dennis Hopper': 1}), 'Judith Anderson': defaultdict(<class 'int'>, {'Richard Wallace': 1}), 'Meghan Ory': defaultdict(<class 'int'>, {'Matthew Hastings': 1}), 'Ocean James': defaultdict(<class 'int'>, {'M. Night Shyamalan': 1}), 'Mike Figueroa': defaultdict(<class 'int'>, {'Brian De Palma': 1}), 'Glenn Campbell': defaultdict(<class 'int'>, {'Oren Peli': 1}), 'Luis Tosar': defaultdict(<class 'int'>, {'Jaume Balagueró': 1, 'Fernando León de Aranoa': 1}), 'Dave Legeno': defaultdict(<class 'int'>, {'Paul Andrew Williams': 1}), 'Teri Polo': defaultdict(<class 'int'>, {'Eric Bross': 1, 'Charles Adelman': 1}), 'James Frain': defaultdict(<class 'int'>, {'Vicky Jewson': 1}), 'David Huddleston': defaultdict(<class 'int'>, {'Peter Hyams': 1}), 'David Raizor': defaultdict(<class 'int'>, {'Mike Mayhall': 1}), 'Mikal Vega': defaultdict(<class 'int'>, {'Stanton Barrett': 1}), 'Julia Jentsch': defaultdict(<class 'int'>, {'Jirí Menzel': 1}), 'Udo Kier': defaultdict(<class 'int'>, {'Fatih Akin': 1}), 'Katy Mixon': defaultdict(<class 'int'>, {'Jeff Nichols': 1, 'Jamie Babbit': 1}), 'Alex Rocco': defaultdict(<class 'int'>, {'Frank LaLoggia': 1}), 'Bill Johnson': defaultdict(<class 'int'>, {'Tobe Hooper': 1}), 'Maria Conchita Alonso': defaultdict(<class 'int'>, {'Gabriela Tagliavini': 1}), 'Bhama': defaultdict(<class 'int'>, {'Shekar': 1}), 'Keith Carradine': defaultdict(<class 'int'>, {'Leonard Farlinger': 1, 'Jonathan Parker': 1}), 'Sara Forestier': defaultdict(<class 'int'>, {'Michel Leclerc': 1}), 'William Kircher': defaultdict(<class 'int'>, {'Robert Sarkies': 1}), 'Brooke Shields': defaultdict(<class 'int'>, {'Randal Kleiser': 1}), 'Royal Dano': defaultdict(<class 'int'>, {'John Huston': 1}), 'Dan Fogler': defaultdict(<class 'int'>, {'': 1}), 'Dana Andrews': defaultdict(<class 'int'>, {'Walter Lang': 1}), 'Mía Maestro': defaultdict(<class 'int'>, {'Carlos Saura': 1, 'Lucrecia Martel': 1}), 'Alex Kendrick': defaultdict(<class 'int'>, {'Andrew Erwin': 1, 'Bill Muir': 1, 'Alex Kendrick': 1}), 'Jaime Camil': defaultdict(<class 'int'>, {'Beto Gómez': 1}), 'Jan Decleir': defaultdict(<class 'int'>, {'Mike van Diem': 1}), 'Rich Pecci': defaultdict(<class 'int'>, {'Todd Solondz': 1}), 'Nicolas Giraud': defaultdict(<class 'int'>, {'René Féret': 1}), 'Simon Yam': defaultdict(<class 'int'>, {'Johnnie To': 1}), 'Oksana Akinshina': defaultdict(<class 'int'>, {'Lukas Moodysson': 1}), 'Jimmy Shergill': defaultdict(<class 'int'>, {'Kabir Sadanand': 1}), 'Katherine McNamara': defaultdict(<class 'int'>, {'Peter DeLuise': 1}), 'Sid Caesar': defaultdict(<class 'int'>, {'Mel Brooks': 1}), 'Nimrat Kaur': defaultdict(<class 'int'>, {'Raja Menon': 1}), 'Fabian Bolin': defaultdict(<class 'int'>, {'Daniel Espinosa': 1}), 'Vidya Balan': defaultdict(<class 'int'>, {'R. Balki': 1, 'Rajkumar Hirani': 1}), 'Yangzom Brauen': defaultdict(<class 'int'>, {'Ivan Engler': 1}), 'Art Hindle': defaultdict(<class 'int'>, {'Bob Clark': 1}), 'Julian Richings': defaultdict(<class 'int'>, {'George A. Romero': 1, 'Robert Eggers': 1}), 'Judith O'Dea': defaultdict(<class 'int'>, {'George A. Romero': 1}), 'Gloria Grahame': defaultdict(<class 'int'>, {'Cecil B. DeMille': 1}), 'Molly C. Quinn': defaultdict(<class 'int'>, {'Duane Journey': 1}), 'Meg Foster': defaultdict(<class 'int'>, {'John Carpenter': 1}), 'Marlee Matlin': defaultdict(<class 'int'>, {'William Arntz': 1}), 'Ellar Coltrane': defaultdict(<class 'int'>, {'Richard Linklater': 1}), 'Angell Conwell': defaultdict(<class 'int'>, {'DJ Pooh': 1}), 'Montgomery Clift': defaultdict(<class 'int'>, {'John Huston': 1, 'Howard Hawks': 1, 'Fred Zinnemann': 1}), 'Geraldine Chaplin': defaultdict(<class 'int'>, {'J.A. Bayona': 1}), 'Ashley Rickards': defaultdict(<class 'int'>, {'Michael Tiddes': 1}), 'Roscoe Lee Browne': defaultdict(<class 'int'>, {'Alfred Hitchcock': 1}), 'Anna Gunn': defaultdict(<class 'int'>, {'Kevin Smith': 1}), 'Stephanie Leonidas': defaultdict(<class 'int'>, {'Dave McKean': 1}), 'Colin Alfredo St. Dic': defaultdict(<class 'int'>, {'Ron Fricke': 1}), 'Kylie Minogue': defaultdict(<class 'int'>, {'Leos Carax': 1}), 'Peter McNamara': defaultdict(<class 'int'>, {'Stefan Schwartz': 1}), 'Lauren Ambrose': defaultdict(<class 'int'>, {'Robert Lee King': 1}), 'Hector Kotsifakis': defaultdict(<class 'int'>, {'George Ramírez Suárez': 1}), 'Noah Emmerich': defaultdict(<class 'int'>, {'David Schwimmer': 1}), 'Rachel Griffiths': defaultdict(<class 'int'>, {'Sam Miller': 1, '': 1}), 'Connor Paolo': defaultdict(<class 'int'>, {'Jim Mickle': 1}), 'Doug Brochu': defaultdict(<class 'int'>, {'': 1}), 'Ed Speleers': defaultdict(<class 'int'>, {'Julian Gilbey': 1}), 'Saad Siddiqui': defaultdict(<class 'int'>, {'Ruba Na

```

```

dda': 1}}, 'Jean Harlow': defaultdict(<class 'int'>, {'Howard Hughes': 1}), 'Lily
Rabe': defaultdict(<class 'int'>, {'Phil Joanou': 1}), 'Robin McLeavy': defaultdi
ct(<class 'int'>, {'Sean Byrne': 1}), 'Kristen Quintrall': defaultdict(<class 'in
t'>, {'Chris Stokes': 1}), 'Eric Mabiuis': defaultdict(<class 'int'>, {'Mark Griff
iths': 1, 'Scott Smith': 1}), 'Rachel Hendrix': defaultdict(<class 'int'>, {'Bruc
e Macdonald': 1, 'Andrew Erwin': 1}), 'Hattie McDaniel': defaultdict(<class 'in
t'>, {'Victor Fleming': 1}), 'Tom Cullen': defaultdict(<class 'int'>, {'Richard R
aymond': 1, 'Andrew Haigh': 1}), 'Keenan Wynn': defaultdict(<class 'int'>, {'Geor
ge Sidney': 1}), 'Kayvan Novak': defaultdict(<class 'int'>, {'Christopher Morri
s': 1}), 'Fernanda Montenegro': defaultdict(<class 'int'>, {'Andrucha Waddingto
n': 1, 'Walter Salles': 1}), 'Gladys Cooper': defaultdict(<class 'int'>, {'Vincen
te Minnelli': 1}), 'Alice Olivia Clarke': defaultdict(<class 'int'>, {'Dagur Kár
i': 1}), 'Brenda Vaccaro': defaultdict(<class 'int'>, {'John Schlesinger': 1}),
'Peter Graves': defaultdict(<class 'int'>, {'Jim Abrahams': 1}), "Ta'Rhonda Jone
s": defaultdict(<class 'int'>, {'': 1}), 'Slim Pickens': defaultdict(<class 'in
t'>, {'Sam Peckinpah': 1}), 'John Forsythe': defaultdict(<class 'int'>, {'Richard
Brooks': 1}), 'Colleen Dewhurst': defaultdict(<class 'int'>, {'Fred Zinnemann':
1}), 'Bernard Cribbins': defaultdict(<class 'int'>, {'Alfred Hitchcock': 1}), 'Jo
e Estevez': defaultdict(<class 'int'>, {'Michael Martin': 1}), 'John Mahoney': de
faultdict(<class 'int'>, {'Edward Burns': 1}), 'Ricardo Darín': defaultdict(<clas
s 'int'>, {'Juan José Campanella': 1, 'Fabián Bielinsky': 1}), 'Gideon Emery': de
faultdict(<class 'int'>, {'Gideon Raff': 1}), 'Ted Raimi': defaultdict(<class 'in
t'>, {'Sam Raimi': 2, 'Bruce Campbell': 1}), 'Wanda Sykes': defaultdict(<class 'i
nt'>, {'Louis C.K.': 1}), 'Elpidia Carrillo': defaultdict(<class 'int'>, {'Salvad
or Carrasco': 1}), 'Otto Jespersen': defaultdict(<class 'int'>, {'André Øvredal':
1}), 'Diahann Carroll': defaultdict(<class 'int'>, {'Warren P. Sonoda': 1}), 'Jan
e Goodall': defaultdict(<class 'int'>, {'Charles Ferguson': 1}), 'Chris Brochu':
defaultdict(<class 'int'>, {'Daniel Petrie Jr.': 1}), 'Murray Bartlett': defaultd
ict(<class 'int'>, {'Eldar Rapaport': 1}), 'Corbin Bleu': defaultdict(<class 'in
t'>, {'Nathan Frankowski': 1}), 'Thomas Bo Larsen': defaultdict(<class 'int'>,
{'Thomas Vinterberg': 1}), 'Armando Riesco': defaultdict(<class 'int'>, {'Alejand
ro Monteverde': 1, 'Zach Braff': 1}), 'Roddy McDowall': defaultdict(<class 'in
t'>, {'Mark L. Lester': 1, 'Don Taylor': 1, 'J. Lee Thompson': 2, 'John Ford':
1}), 'François Bégaudeau': defaultdict(<class 'int'>, {'Laurent Cantet': 1}), 'Le
e Marvin': defaultdict(<class 'int'>, {'John Ford': 2, 'John Boorman': 1}), 'Cata
lina Sandino Moreno': defaultdict(<class 'int'>, {'Joshua Marston': 1}), 'Honor B
lackman': defaultdict(<class 'int'>, {'Guy Hamilton': 1}), 'Sarah Gadon': default
dict(<class 'int'>, {'Brandon Cronenberg': 1}), 'Donna Reed': defaultdict(<class
'int'>, {'Frank Capra': 1}), 'Sprague Grayden': defaultdict(<class 'int'>, {'Tod
Williams': 1}), 'Clu Gulager': defaultdict(<class 'int'>, {'Jack Sholder': 1}),
'Rhys Wakefield': defaultdict(<class 'int'>, {'James DeMonaco': 1}), 'Danielle Ko
tch': defaultdict(<class 'int'>, {'Scott Derrickson': 1}), "Nancy O'Dell": default
dict(<class 'int'>, {'David Raynr': 1}), 'Burl Ives': defaultdict(<class 'int'>,
{'Richard Brooks': 1}), 'Linda Harrison': defaultdict(<class 'int'>, {'Ted Post':
1}), 'Veronica Taylor': defaultdict(<class 'int'>, {'Kunihiko Yuyama': 1}), 'Ed L
auter': defaultdict(<class 'int'>, {'Alfred Hitchcock': 1}), 'Fred MacMurray': de
faultdict(<class 'int'>, {'Billy Wilder': 1}), 'Tamara Tomanova': defaultdict(<c
lass 'int'>, {'Alfred Hitchcock': 1}), 'Shirley Jones': defaultdict(<class 'in
t'>, {'Richard Brooks': 1}), 'Maximilian Schell': defaultdict(<class 'int'>, {'St
anley Kramer': 1}), 'Matthew Senreich': defaultdict(<class 'int'>, {'': 1}), 'Sea
mus Davey-Fitzpatrick': defaultdict(<class 'int'>, {'Richard Linklater': 1}), 'Jo
hn Astin': defaultdict(<class 'int'>, {'Christopher Leitch': 1}), 'Angus Scrimm':
defaultdict(<class 'int'>, {'Don Coscarelli': 1}), 'John Abraham': defaultdict(<c
lass 'int'>, {'Deepa Mehta': 1}), 'Callie Thorne': defaultdict(<class 'int'>, {'P
eter M. Cohen': 1}), 'Peter Ostrum': defaultdict(<class 'int'>, {'Mel Stuart':
1}), 'Kristy Swanson': defaultdict(<class 'int'>, {'Tom Schulman': 1}), 'Greta Ge
rwig': defaultdict(<class 'int'>, {'Noah Baumbach': 1, 'Whit Stillman': 1, 'Jay D
uplass': 1}), 'Eddie Griffin': defaultdict(<class 'int'>, {'George Gallo': 1, 'Da
ve Meyers': 1, 'Nick Love': 1}), 'Robyn Lively': defaultdict(<class 'int'>, {'Dav
id Nixon': 1}), 'Kevin Pollak': defaultdict(<class 'int'>, {'Enrique Begne': 1,

```

```

'Rod Lurie': 1}}, 'William Morgan Sheppard': defaultdict(<class 'int'>, {'Michael
Landon Jr.': 1}}, 'Michael Gross': defaultdict(<class 'int'>, {'Tim Heidecker':
1}}, 'Susan Hampshire': defaultdict(<class 'int'>, {'': 1}}, 'Alicja Bachleda': d
efaultdict(<class 'int'>, {'Marco Kreuzpaintner': 1}}, 'Johnny Pacar': defaultdic
t(<class 'int'>, {'Kate Connor': 1, 'Matthew R. Anderson': 1}}, 'Matt Cohen': def
aultdict(<class 'int'>, {'Deon Taylor': 1}}, 'Neil Hopkins': defaultdict(<class
'int'>, {'Philip Zlotorynski': 1}}, 'Tom Everett Scott': defaultdict(<class 'in
t'>, {'Francesca Gregorini': 1, 'W.D. Hogan': 1}}, 'John Savage': defaultdict(<cl
ass 'int'>, {'Marcio Garcia': 1}}, 'French Stewart': defaultdict(<class 'int'>,
{'Craig Moss': 1}}, 'Everett McGill': defaultdict(<class 'int'>, {'Monte Hellma
n': 1}}, 'Ann-Margret': defaultdict(<class 'int'>, {'Frank Capra': 1}}, 'Anthony
L. Fernandez': defaultdict(<class 'int'>, {'Isaac Florentine': 1}}, 'Dyan Canno
n': defaultdict(<class 'int'>, {'Susan Seidelman': 1}}, 'Olivia Rose Keegan': def
aultdict(<class 'int'>, {'Michael Polish': 1}}, 'Greg Crowe': defaultdict(<class
'int'>, {'Heidi Ewing': 1}}, 'Cécile De France': defaultdict(<class 'int'>, {'Ale
xandre Aja': 1}}, 'Maeve Dermody': defaultdict(<class 'int'>, {'Leon Ford': 1}},
'Isaac Hayes': defaultdict(<class 'int'>, {'Craig Brewer': 1}}, 'Nehemiah Persof
f': defaultdict(<class 'int'>, {'Billy Wilder': 1}}, 'Margaret Hamilton': default
dict(<class 'int'>, {'Victor Fleming': 1}}, 'Megan Park': defaultdict(<class 'in
t'>, {'George A. Romero': 1}}, 'Teri Garr': defaultdict(<class 'int'>, {'Carroll
Ballard': 1}}, 'Bhavkhandan Singh Rakhra': defaultdict(<class 'int'>, {'Rohit Jug
raj': 1}}, 'Iben Hjejle': defaultdict(<class 'int'>, {'Thorbjørn Christoffersen':
1}}, 'Rijk de Gooyer': defaultdict(<class 'int'>, {'Alex van Warmerdam': 1}}, 'Sp
encer Tracy': defaultdict(<class 'int'>, {'Victor Fleming': 1}}, 'Judie Aronson':
defaultdict(<class 'int'>, {'Joseph Zito': 1}}, 'Joanna Kulig': defaultdict(<clas
s 'int'>, {'Pawel Pawlikowski': 1}}, 'Rupert Graves': defaultdict(<class 'int'>,
{'James Ivory': 1}}, 'Gerry Lopez': defaultdict(<class 'int'>, {'Stacy Peralta':
1}}, 'Karra Elejalde': defaultdict(<class 'int'>, {'Nacho Vigalondo': 1}}, 'Jack
Kao': defaultdict(<class 'int'>, {'Hao Ning': 1}}, 'Brianna Brown': defaultdict(<
class 'int'>, {'Tony Giglio': 1}}, 'Joe Cross': defaultdict(<class 'int'>, {'Joe
Cross': 1}}, 'Essence Atkins': defaultdict(<class 'int'>, {'Michael Tiddes': 1,
'David M. Matthews': 1}}, 'Barack Obama': defaultdict(<class 'int'>, {"Dinesh D'S
ouza": 1}}, 'Tom Atkins': defaultdict(<class 'int'>, {'Tommy Lee Wallace': 1}},
'Patricia Neal': defaultdict(<class 'int'>, {'Martin Ritt': 1}}, 'David Jason Per
ez': defaultdict(<class 'int'>, {'Leslie Small': 1}}, 'Hanno Pöschl': defaultdict
(<class 'int'>, {'Richard Linklater': 1}}, 'Patrick Huard': defaultdict(<class 'i
nt'>, {'Éric Tessier': 1}}, 'Joan Chen': defaultdict(<class 'int'>, {'Alice Wu':
1}}, 'Satish Kaushik': defaultdict(<class 'int'>, {'Sarah Gavron': 1}}, 'Bill Mah
er': defaultdict(<class 'int'>, {'Larry Charles': 1}}, 'Dustin Lance Black': defa
ultdict(<class 'int'>, {'Reed Cowan': 1}}, 'Benno Fürmann': defaultdict(<class 'i
nt'>, {'Stefan Ruzowitzky': 1}}, 'Leonor Varela': defaultdict(<class 'int'>, {'Al
ex Rivera': 1}}, 'Ashley Scott': defaultdict(<class 'int'>, {'John Murlowski':
1}}, "Anthony 'Citric' Campos": defaultdict(<class 'int'>, {'Mike Judge': 1}}, 'S
imon Merrells': defaultdict(<class 'int'>, {'Zackary Adler': 1}}, 'Esther William
s': defaultdict(<class 'int'>, {'George Sidney': 1}}, 'Jossara Jinaro': defaultdi
ct(<class 'int'>, {'Carmen Marron': 1}}, 'Agnes Moorehead': defaultdict(<class 'i
nt'>, {'George Sidney': 1}}, 'Leopold Stokowski': defaultdict(<class 'int'>, {'Ja
mes Algar': 1}}, 'Nicole Smolen': defaultdict(<class 'int'>, {'Jaco Booyens':
1}}, 'Richard Brooker': defaultdict(<class 'int'>, {'Steve Miner': 1}}, 'Tiffany
Helm': defaultdict(<class 'int'>, {'Danny Steinmann': 1}}, 'Brian Austin Green':
defaultdict(<class 'int'>, {'Hayley Cloake': 1}}, 'Jessica Tandy': defaultdict(<c
lass 'int'>, {'Tay Garnett': 1}}, 'Saeed Jaffrey': defaultdict(<class 'int'>, {'H
armage Singh Kalirai': 1, 'Stephen Frears': 1}}, 'Rachel Skarsten': defaultdict(<
class 'int'>, {'Jon Knautz': 1}}, 'Myrna Loy': defaultdict(<class 'int'>, {'Willi
am Wyler': 1}}, 'Jørgen Langhelle': defaultdict(<class 'int'>, {'Petter Næss':
1}}, 'Michael Derek': defaultdict(<class 'int'>, {'Robert Fontaine': 1}}, 'Manuel
a Velasco': defaultdict(<class 'int'>, {'Jaume Balagueró': 1}}, 'Travis Myers': d
efaultdict(<class 'int'>, {'Dan Perri': 1}}, 'Walter Hendrix III': defaultdict(<c
lass 'int'>, {'Neil Mcenery-West': 1}}, 'James Ransone': defaultdict(<class 'in
t'>, {"Anthony O'Brien": 1}}, 'Phoebe Legere': defaultdict(<class 'int'>, {'Micha

```

```

el Herz': 1}}, "Carroll O'Connor": defaultdict(<class 'int'>, {'': 1}}, 'Maika Mo
nroe': defaultdict(<class 'int'>, {'David Robert Mitchell': 1}}, 'Vernon Wells':
defaultdict(<class 'int'>, {'George Miller': 1}}, 'Alicia Goranson': defaultdict
(<class 'int'>, {'Kimberly Peirce': 1}}, 'Eric Sheffer Stevens': defaultdict(<cla
ss 'int'>, {'Chris Kentis': 1}}, 'Ben Davies': defaultdict(<class 'int'>, {'Alex
Kendrick': 1}}, 'Hedy Lamarr': defaultdict(<class 'int'>, {'Jack Conway': 1}}, 'M
ichel Robin': defaultdict(<class 'int'>, {'Sylvain Chomet': 1}}, 'Michael Greeye
s': defaultdict(<class 'int'>, {'Chris Eyre': 1}}, 'Vernon Dobtcheff': defaultdic
t(<class 'int'>, {'Richard Linklater': 1}}, 'Adriana Barraza': defaultdict(<class
'int'>, {'Alejandro G. Iñárritu': 1}}, 'Nicholas Aaron': defaultdict(<class 'in
t'>, {'Kevin Macdonald': 1}}, 'Tyrone Power': defaultdict(<class 'int'>, {'Henry
King': 1}}, 'Najarra Townsend': defaultdict(<class 'int'>, {'Miranda July': 1}},
'Vanessa Lengies': defaultdict(<class 'int'>, {'Max Joseph': 1}}, 'Henry Silva':
defaultdict(<class 'int'>, {'Jim Jarmusch': 1}}, 'Kathleen Wilhoite': defaultdict
(<class 'int'>, {'Kevin Tenney': 1}}, 'Stephen Colbert': defaultdict(<class 'in
t'>, {'': 1}}, 'Sook-Yin Lee': defaultdict(<class 'int'>, {'John Cameron Mitchel
l': 1}}, 'Ari Folman': defaultdict(<class 'int'>, {'Ari Folman': 1}}, 'Noah Danb
y': defaultdict(<class 'int'>, {'Gary Rogers': 1}}, 'John F. Kennedy': defaultdic
t(<class 'int'>, {'David Singleton': 1}}, 'Danny Aiello': defaultdict(<class 'in
t'>, {'Bob Giraldi': 1}}, 'Alanna Ubach': defaultdict(<class 'int'>, {'Jill Sprec
her': 1, 'Rob Bowman': 1}}, 'Matt Bennett': defaultdict(<class 'int'>, {'Huck Bot
ko': 1}}, 'Jess Weixler': defaultdict(<class 'int'>, {'Mitchell Lichtenstein':
1}}, 'Jeffrey Falcon': defaultdict(<class 'int'>, {'Lance Mungia': 1}}, 'Karisma
Kapoor': defaultdict(<class 'int'>, {'Kundan Shah': 1, 'Khalid Mohamed': 1}}, 'Pa
ul Kaye': defaultdict(<class 'int'>, {'Michael Dowse': 1}}, 'Merritt Wever': defa
ultdict(<class 'int'>, {'Noah Buschel': 1}}, 'Brad William Henke': defaultdict(<c
lass 'int'>, {'Laurie Collyer': 1}}, 'Jordi Vilasuso': defaultdict(<class 'int'>,
{'Aaron Hann': 1}}, 'Rapulana Seiphemo': defaultdict(<class 'int'>, {'Ralph Zima
n': 1}}, 'Mackenzie Gray': defaultdict(<class 'int'>, {'Colin Minihan': 1}}, 'Pet
er McQuinn': defaultdict(<class 'int'>, {'Conor McMahon': 1}}, 'Daniel Baldwin':
defaultdict(<class 'int'>, {'Chris Shadley': 1}}, 'Alice Cooper': defaultdict(<cl
ass 'int'>, {'Simon Napier-Bell': 1}}, 'Paul Sheehan': defaultdict(<class 'int'>,
{'Simon Yin': 1}}, 'Steven Weber': defaultdict(<class 'int'>, {'': 1}}, 'Rita Hay
worth': defaultdict(<class 'int'>, {'Orson Welles': 1}}, 'Mircea Monroe': default
dict(<class 'int'>, {'Dave Payne': 1}}, 'Sunil Narkar': defaultdict(<class 'in
t'>, {'Brad Copeland': 1}}, 'Kate Maberly': defaultdict(<class 'int'>, {'Paul Bun
nell': 1}}, 'Jon Mack': defaultdict(<class 'int'>, {'James Cox': 1}}, 'Buster Kea
ton': defaultdict(<class 'int'>, {'Richard Lester': 1}}, 'Trenton Rostedt': defau
ltdict(<class 'int'>, {'Scott Dow': 1}}, 'Melissa Bolona': defaultdict(<class 'in
t'>, {'Jerry Dugan': 1}}, 'Paul McGillion': defaultdict(<class 'int'>, {'William
Kaufman': 1}}, 'Khalid Abdalla': defaultdict(<class 'int'>, {'Jehane Noujaim':
1}}, 'Karl E. Landler': defaultdict(<class 'int'>, {'Pascal Arnold': 1}}, 'Lauren
Gottlieb': defaultdict(<class 'int'>, {'Remo': 1}}, 'Paul McGill': defaultdict(<c
lass 'int'>, {'James Marsh': 1}}, 'Siobhan Marshall': defaultdict(<class 'int'>,
{'John Laing': 1}}, 'Logan Browning': defaultdict(<class 'int'>, {'Jamal Hill':
1}}, 'Caleb Landry Jones': defaultdict(<class 'int'>, {'Daniel Stamm': 1}}, 'Dami
án Alcázar': defaultdict(<class 'int'>, {'Carlos Carrera': 1}}, 'Donna Murphy': d
efaultdict(<class 'int'>, {'Vera Farmiga': 1}}, 'Hansika Motwani': defaultdict(<c
lass 'int'>, {'Vijay Chandar': 1}}, 'Tshewang Dendup': defaultdict(<class 'int'>,
{'Khyentse Norbu': 1}}, 'Humphrey Bogart': defaultdict(<class 'int'>, {'Richard B
rooks': 1, 'Michael Curtiz': 1}}, 'George Newbern': defaultdict(<class 'int'>,
{'Tony Krantz': 1}}, 'Kimberley Crossman': defaultdict(<class 'int'>, {'Mitchell
Altieri': 1}}, 'Christian Keyes': defaultdict(<class 'int'>, {'Corey Grant': 1}},
'James Duval': defaultdict(<class 'int'>, {'Lucky McKee': 1}}, 'Matthew Alan': de
faultdict(<class 'int'>, {'David Hunt': 1}}, 'Sarah Butler': defaultdict(<class
'int'>, {'Steven R. Monroe': 1}}, 'Ally Walker': defaultdict(<class 'int'>, {'Mar
k Illsley': 1}}, 'Nawazuddin Siddiqui': defaultdict(<class 'int'>, {'Ritesh Batr
a': 1}}, 'Jean Arthur': defaultdict(<class 'int'>, {'Frank Capra': 1}}, 'Harry Ca
rey Jr.': defaultdict(<class 'int'>, {'John Ford': 1}}, 'Shawnee Smith': defaultd
ict(<class 'int'>, {'Brad J. Silverman': 1}}, 'Yasmine Al Massri': defaultdict(<c

```

```

class 'int'), {'Nadine Labaki': 1}), 'Ohad Knoller': defaultdict(<class 'int'>,
{'Eytan Fox': 1}), 'Annabelle Wallis': defaultdict(<class 'int'>, {'Romesh Sharm
a': 1}), 'Dustin Ingram': defaultdict(<class 'int'>, {'Travis Zariwny': 1}), 'Lew
Temple': defaultdict(<class 'int'>, {'Adrienne Shelly': 1}), 'William Baldwin': d
efaultdict(<class 'int'>, {'Noah Baumbach': 1}), 'Scott Cohen': defaultdict(<clas
s 'int'>, {'Charles Herman-Wurmfeld': 1}), 'Matthew Ziff': defaultdict(<class 'in
t'>, {'John Stockwell': 1, 'Johnny Remo': 1, 'Travis Romero': 1}), 'Norman Lloy
d': defaultdict(<class 'int'>, {'Alfred Hitchcock': 1}), 'Ami Ayalon': defaultdic
t(<class 'int'>, {'Dror Moreh': 1}), 'Beau Bridges': defaultdict(<class 'int'>,
{'Rebecca Miller': 1}), 'Donald Glover': defaultdict(<class 'int'>, {'Maggie Care
y': 1}), 'Ward G. Smith': defaultdict(<class 'int'>, {'Simeon Rice': 1}), 'Paul G
uilfoyle': defaultdict(<class 'int'>, {'Brad Anderson': 1}), 'Jessy Schram': defa
ultdict(<class 'int'>, {'Jeff Garlin': 1}), 'Joel David Moore': defaultdict(<clas
s 'int'>, {'Adam Green': 1}), 'Paulette Goddard': defaultdict(<class 'int'>, {'Ch
arles Chaplin': 1}), 'Jay Mohr': defaultdict(<class 'int'>, {'Adam Carolla': 1}),
'Brian Lee Franklin': defaultdict(<class 'int'>, {'John Sayles': 1}), 'Angela Sar
afyan': defaultdict(<class 'int'>, {'Jaime Zavallos': 1}), "Morgana O'Reilly": de
faultdict(<class 'int'>, {'Gerard Johnstone': 1}), 'Lee Scott': defaultdict(<clas
s 'int'>, {'Robert Greenwald': 1}), 'Nicole Muñoz': defaultdict(<class 'int'>,
{'David Ray': 1}), 'Monica Barladeanu': defaultdict(<class 'int'>, {'Nae Caranfi
l': 1}), 'Ashanti': defaultdict(<class 'int'>, {'David Winning': 1}), 'Brighton S
harbino': defaultdict(<class 'int'>, {'Frank Lotito': 1, 'E.L. Katz': 1}), 'Katri
na Law': defaultdict(<class 'int'>, {'Timothy Woodward Jr.': 1}), 'Timothy Hutto
n': defaultdict(<class 'int'>, {'Tara Subkoff': 1}), 'Adriana Caselotti': default
dict(<class 'int'>, {'William Cottrell': 1}), 'Brigitte Bardot': defaultdict(<cla
ss 'int'>, {'Edward Dmytryk': 1}), 'Zak Penn': defaultdict(<class 'int'>, {'Zak P
enn': 1}), 'Camren Bicondova': defaultdict(<class 'int'>, {'Jon Knautz': 1}), 'Je
ssica Szohr': defaultdict(<class 'int'>, {'David Worth': 1}), 'Burgess Meredith':
defaultdict(<class 'int'>, {'Leslie H. Martinson': 1}), 'Karl Malden': defaultdic
t(<class 'int'>, {'': 1}), 'Ulrich Thomsen': defaultdict(<class 'int'>, {'Thomas
Vinterberg': 1}), 'Long Nguyen': defaultdict(<class 'int'>, {'Ham Tran': 1}), 'Ja
mes Morrison': defaultdict(<class 'int'>, {'': 1}), 'Elina Abai Kyzy': defaultdic
t(<class 'int'>, {'Sadyk Sher-Niyaz': 1}), 'Maury Chaykin': defaultdict(<class 'i
nt'>, {'Paul Donovan': 1}), 'Noureen DeWulf': defaultdict(<class 'int'>, {'Craig
Moss': 1}), 'Betsy Palmer': defaultdict(<class 'int'>, {'Steve Miner': 1}), 'Ray
Milland': defaultdict(<class 'int'>, {'Billy Wilder': 1}), 'Vic Mignogna': defaul
tdict(<class 'int'>, {'Deryck Broom': 1}), 'Geoffrey Arend': defaultdict(<class
'int'>, {'Jay Chandrasekhar': 1}), 'Zuhair Haddad': defaultdict(<class 'int'>,
{'Giovanni Zelko': 1}), 'Ximena Ayala': defaultdict(<class 'int'>, {'Claudia Sain
te-Luce': 1}), 'Jerry Mathers': defaultdict(<class 'int'>, {'Alfred Hitchcock':
1}), 'Spring Byington': defaultdict(<class 'int'>, {'Walter Lang': 1}), 'Alex Hus
zar': defaultdict(<class 'int'>, {'Anna Muylaert': 1}), 'Jason Marsden': defaulttd
ict(<class 'int'>, {'Steve Taylor': 1}), 'Lemmy': defaultdict(<class 'int'>, {'Ku
rt Voss': 1}), '': defaultdict(<class 'int'>, {'Léa Pool': 1, 'Harry Gantz': 1,
'U. Roberto Romano': 1, 'Pan Nalin': 1, 'Amal Al-Agroobi': 1, 'Andrew Berends':
1, 'Jem Cohen': 1}), 'Déborah Révy': defaultdict(<class 'int'>, {'Laurent Bouhni
k': 1}), 'Nataniel Sánchez': defaultdict(<class 'int'>, {'Tom Sanchez': 1}), 'Ric
h Ceraulo': defaultdict(<class 'int'>, {'Adam Jay Epstein': 1}), 'Errol Flynn': d
efaultdict(<class 'int'>, {'Michael Curtiz': 1}), 'Christie Burke': defaultdict(<
class 'int'>, {'Brenton Spencer': 1}), 'Redd Foxx': defaultdict(<class 'int'>,
{'Ossie Davis': 1}), "Seregon O'Dassey": defaultdict(<class 'int'>, {'Dylan Ban
k': 1}), 'Richard Cawthorne': defaultdict(<class 'int'>, {'': 1}), 'Terry Kiser':
defaultdict(<class 'int'>, {'Jeff Burr': 1}), 'Sarah Murphy-Dyson': defaultdict(<
class 'int'>, {'Guy Maddin': 1}), 'Sean Cameron Michael': defaultdict(<class 'in
t'>, {'Regardt van den Bergh': 1}), 'Chris Gauthier': defaultdict(<class 'int'>,
{'Panos Cosmatos': 1}), 'Iko Uwais': defaultdict(<class 'int'>, {'Gareth Evans':
1}), 'Piero Maggì': defaultdict(<class 'int'>, {'Joe Marino': 1}), 'Talia Zucke
r': defaultdict(<class 'int'>, {'Joel Anderson': 1}), 'Christopher Abbott': defau
ltdict(<class 'int'>, {'Mona Fastvold': 1, 'Sean Durkin': 1}), 'Shelley Hennig':
defaultdict(<class 'int'>, {'Levan Gabriadze': 1}), 'Ursula Andress': defaultdict

```

```
(<class 'int'>, {'Terence Young': 1}), 'Andrew Robinson': defaultdict(<class 'int'>, {'Clive Barker': 1}), 'Benjamin A. Onyango': defaultdict(<class 'int'>, {'Harold Cronk': 1}), 'Hiroshi Abe': defaultdict(<class 'int'>, {'Takao Okawara': 1}), 'Fernanda Andrade': defaultdict(<class 'int'>, {'William Brent Bell': 1}), 'James Drury': defaultdict(<class 'int'>, {'Robert D. Webb': 1}), 'Billy West': defaultdict(<class 'int'>, {'Davis Guggenheim': 1}), 'James Brown': defaultdict(<class 'int'>, {'Allan Dwan': 1}), 'Mick Jagger': defaultdict(<class 'int'>, {'Martin Scorsese': 1}), 'Juliet Stevenson': defaultdict(<class 'int'>, {'Andrew Stegga ll': 1}), 'W. Earl Brown': defaultdict(<class 'int'>, {'Ben Lewin': 1}), 'Michael Pollan': defaultdict(<class 'int'>, {'Robert Kenner': 1}), 'Gaby Hoffmann': defaultdict(<class 'int'>, {'Gillian Robespierre': 1}), 'Sheryl Crow': defaultdict(<class 'int'>, {'Morgan Neville': 1}), 'Kevin Corrigan': defaultdict(<class 'int'>, {'Nicole Holofcener': 1, 'Steve Buscemi': 1, 'Jack Perez': 1}), 'Sonja Sohn': defaultdict(<class 'int'>, {'Marc Levin': 1}), 'Trey Parker': defaultdict(<class 'int'>, {'Trey Parker': 1}), 'Shahir Kabaha': defaultdict(<class 'int'>, {'Scandar Copti': 1}), 'Steven Yeun': defaultdict(<class 'int'>, {'Mike Cahill': 1}), 'Shiri Appleby': defaultdict(<class 'int'>, {'Efram Potelle': 1}), 'Anwar Congo': defaultdict(<class 'int'>, {'Joshua Oppenheimer': 1}), 'Alex Gibney': defaultdict(<class 'int'>, {'Alex Gibney': 1}), 'Pelé': defaultdict(<class 'int'>, {'Paul Crowder': 1}), 'Kumar Gaurav': defaultdict(<class 'int'>, {'Rohit Jagessar': 1}), 'Josh Swanson': defaultdict(<class 'int'>, {'Anthony Powell': 1}), 'G.W. Krauss': defaultdict(<class 'int'>, {'Kief Davidson': 1}), 'Emraan Hashmi': defaultdict(<class 'int'>, {'Vivek Agnihotri': 1}), 'Lena Dunham': defaultdict(<class 'int'>, {'Ti West': 2, 'Daniel Schechter': 1, 'Lena Dunham': 1}), 'Harvey Fierstein': defaultdict(<class 'int'>, {'John Hamburg': 1}), 'Patrick Renna': defaultdict(<class 'int'>, {'Nickolas Perry': 1}), 'Yul Vazquez': defaultdict(<class 'int'>, {'Dan Zukovic': 1}), 'Will Rothhaar': defaultdict(<class 'int'>, {'Becky Smith': 1}), 'Jordana Spiro': defaultdict(<class 'int'>, {'Eric Nicholas': 1}), 'Nora Zehetner': defaultdict(<class 'int'>, {'Benjamin Dickinson': 1}), 'Frances Bay': defaultdict(<class 'int'>, {'Darren Stein': 1}), 'Scott Levy': defaultdict(<class 'int'>, {'A. Raven Cruz': 1}), 'Kenny Johnston': defaultdict(<class 'int'>, {'Amat Escalante': 1}), 'Rasheed Naz': defaultdict(<class 'int'>, {'Wajahat Rauf': 1}), 'Ilene Graff': defaultdict(<class 'int'>, {'Katherine Brooks': 1}), 'Jason Ritter': defaultdict(<class 'int'>, {'David Cross': 1}), 'Nicole Randall Johnson': defaultdict(<class 'int'>, {'Randall Rubin': 1}), 'Randy Wayne': defaultdict(<class 'int'>, {'Robert Bennett': 1, 'Brian Baugh': 1}), 'Russell Hodgkinson': defaultdict(<class 'int'>, {'Kevin Hamedani': 1}), 'Maureen McCormick': defaultdict(<class 'int'>, {'David DeCoteau': 1}), 'Álex Angulo': defaultdict(<class 'int'>, {'Mikel Rueda': 1}), 'Katie Aselton': defaultdict(<class 'int'>, {'Katie Aselton': 1}), 'David Oakes': defaultdict(<class 'int'>, {'Robert Heath': 1}), 'Magi Avila': defaultdict(<class 'int'>, {'D. Stevens': 1}), 'Boyd Banks': defaultdict(<class 'int'>, {'Jeffrey St. Jules': 1}), 'Angelina Armani': defaultdict(<class 'int'>, {'Robert Hall': 1}), 'Jason K. Wixom': defaultdict(<class 'int'>, {'K. King': 1}), 'Justin Baldoni': defaultdict(<class 'int'>, {'Jonathan Meyers': 1}), 'Monique Candelaria': defaultdict(<class 'int'>, {'Blair Erickson': 1}), 'Eric Lloyd': defaultdict(<class 'int'>, {'': 1}), 'Kimberly Williams-Paisley': defaultdict(<class 'int'>, {'Alison Burnett': 1}), 'Alma Martinez': defaultdict(<class 'int'>, {'Sanjay Rawal': 1}), 'Justin Kirk': defaultdict(<class 'int'>, {'Kat Coiro': 1}), 'Anamaria Marinca': defaultdict(<class 'int'>, {'Cristian Mungiu': 1}), 'Odessa Rae': defaultdict(<class 'int'>, {'David Slade': 1}), 'Sarah Kazemy': defaultdict(<class 'int'>, {'Maryam Keshavarz': 1}), 'Ahna O'Reilly': defaultdict(<class 'int'>, {'Ryan Coogler': 1}), 'Sam Edwards': defaultdict(<class 'int'>, {'James Algar': 1}), 'John Enos III': defaultdict(<class 'int'>, {'Charles Herman-Wurmfeld': 1}), 'Stana Roumillac': defaultdict(<class 'int'>, {'Mariette Monpierre': 1}), 'James Wolk': defaultdict(<class 'int'>, {'Jamie Travis': 1}), 'Gavin MacLeod': defaultdict(<class 'int'>, {'Rich Christiano': 1}), 'Lorraine Stanley': defaultdict(<class 'int'>, {'Paul Andrew Williams': 1}), 'Tommy O'Reilly': defaultdict(<class 'int'>, {'Natalie Bible': 1}), 'Shahab Hosseini': defaultdict(<class 'int'>, {'Asghar Farhad': 1}), 'Jo Hartley': defaultdict(<class 'int'>, {'Justin Molotnikov': 1}), 'Allison Dean': defaultdict(<class 'int'>, {'Victor Nunez': 1}), 'Victor Rasuk': defa
```



```

ulldict(<class 'int'>, {'Peter Sollett': 1}), 'Louise Brooks': defaultdict(<class
'int'>, {'Georg Wilhelm Pabst': 1}), 'Norma Aleandro': defaultdict(<class 'int'>,
{'Jorge Gaggero': 1}), 'Zachary Knighton': defaultdict(<class 'int'>, {'Michael B
urke': 1}), 'Shane Dawson': defaultdict(<class 'int'>, {'Shane Dawson': 1}), 'Cor
bin Allred': defaultdict(<class 'int'>, {'Ryan Little': 1}), 'Taylor Pigeon': def
aultdict(<class 'int'>, {'Clark Baker': 1}), 'Scott Helvenston': defaultdict(<cla
ss 'int'>, {'Robert Greenwald': 1}), 'Isaac C. Singleton Jr.': defaultdict(<class
'int'>, {'Leslie Small': 1}), 'Jean-Louis Trintignant': defaultdict(<class 'in
t'>, {'Bernardo Bertolucci': 1}), 'Raymond J. Barry': defaultdict(<class 'int'>,
{'Neil Burger': 1}), 'Luke Grimes': defaultdict(<class 'int'>, {'Jonathan Levin
e': 1}), 'Dichen Lachman': defaultdict(<class 'int'>, {'Christopher Hutson': 1}),
'William Gates': defaultdict(<class 'int'>, {'Steve James': 1}), 'Kevin Scott Ric
hardson': defaultdict(<class 'int'>, {'David LaChapelle': 1}), 'Adam LeFevre': de
faultdict(<class 'int'>, {'Michael Cuesta': 1}), 'Laura Fraser': defaultdict(<cla
ss 'int'>, {'Caryn Waechter': 1}), 'Irene Bedard': defaultdict(<class 'int'>, {'J
ennifer Wynne Farmer': 1}), 'Hill Harper': defaultdict(<class 'int'>, {'Christoph
er Scott Cherot': 1}), 'Trey Loney': defaultdict(<class 'int'>, {'Randy Moore':
1}), 'Richard Curtis': defaultdict(<class 'int'>, {'Chris Atkins': 1}), 'Leven Ra
mbin': defaultdict(<class 'int'>, {'Anna Mastro': 1}), 'Ginger Rogers': defaultdi
ct(<class 'int'>, {'Mark Sandrich': 1, 'Lloyd Bacon': 1}), 'Heather Donahue': def
aultdict(<class 'int'>, {'Daniel Myrick': 1}), 'Joe Cocker': defaultdict(<class
'int'>, {'Michael Wadleigh': 1}), 'David Zucker': defaultdict(<class 'int'>, {'Jo
hn Landis': 1}), 'Stacy Keach': defaultdict(<class 'int'>, {'Jon Gunn': 1, 'Matt
Cimber': 1}), 'Kendyl Joi': defaultdict(<class 'int'>, {'Gene Teigland': 1}), 'Al
iou Touré': defaultdict(<class 'int'>, {'Johanna Schwartz': 1}), 'Karen Alexande
r': defaultdict(<class 'int'>, {'William Gazecki': 1}), 'Russell Wong': defaultdi
ct(<class 'int'>, {'Lance McDaniel': 1}), 'Paula Garcés': defaultdict(<class 'in
t'>, {'Michael Walker': 1}), 'Paul McCartney': defaultdict(<class 'int'>, {'Richa
rd Lester': 1}), 'Dario Nozic Serini': defaultdict(<class 'int'>, {'Blaz Zavrnsni
k': 1}), 'Kirk Cameron': defaultdict(<class 'int'>, {'Alex Kendrick': 1}), 'Bhole
Baba': defaultdict(<class 'int'>, {'Pan Nalin': 1}), 'Frances Bavier': defaultdic
t(<class 'int'>, {'Joe Camp': 1}), 'Blanchard Ryan': defaultdict(<class 'int'>,
{'Chris Kentis': 1}), 'Woody Strode': defaultdict(<class 'int'>, {"John 'Bud' Car
dos": 1}), 'Heather Berman': defaultdict(<class 'int'>, {'Marilyn Agrelo': 1}),
'Ken Burns': defaultdict(<class 'int'>, {'Patrick Creadon': 1}), 'Terry Funk': de
faultdict(<class 'int'>, {'Barry W. Blaustein': 1}), 'Kirby Heyborne': defaultdic
t(<class 'int'>, {'Kurt Hale': 2}), 'Marina Golbahari': defaultdict(<class 'in
t'>, {'Siddiq Barmak': 1}), 'Rachel Dratch': defaultdict(<class 'int'>, {'Joseph
Dorman': 1}), 'Rachel True': defaultdict(<class 'int'>, {'Greg Harrison': 1}), 'S
asha Alexander': defaultdict(<class 'int'>, {'Michael Polish': 1}), 'Edie Falco':
defaultdict(<class 'int'>, {'Morgan J. Freeman': 1}), 'Monica Calhoun': defaultdi
ct(<class 'int'>, {'Neema Barnette': 1}), 'David Mamet': defaultdict(<class 'in
t'>, {'Molly Bernstein': 1}), 'Takashi Shimura': defaultdict(<class 'int'>, {'Aki
ra Kurosawa': 1}), 'Tommy Sheppard': defaultdict(<class 'int'>, {'Marius A. Marke
vicius': 1}), 'Monique Gabriela Curnen': defaultdict(<class 'int'>, {'Justin Li
n': 1}), 'Haley Ramm': defaultdict(<class 'int'>, {'Quentin Dupieux': 1}), 'Tori
Spelling': defaultdict(<class 'int'>, {'C. Jay Cox': 1}), 'Bo Huang': defaultdict
(<class 'int'>, {'Hao Ning': 1}), 'Brian Eno': defaultdict(<class 'int'>, {'Steph
en Kijak': 1}), 'Daston Kalili': defaultdict(<class 'int'>, {'Daston Kalili':
1}), 'Carol Channing': defaultdict(<class 'int'>, {'William Gazecki': 1}), 'Vince
nt Chimato': defaultdict(<class 'int'>, {'Michael Hoffman Jr.': 1}), 'Emma Caulfi
eld': defaultdict(<class 'int'>, {'Patrick Gilles': 1}), 'Mike Beckingham': defau
ltdict(<class 'int'>, {'Georgia Hilton': 1}), 'Shaun Sipos': defaultdict(<class
'int'>, {'Jack Heller': 1}), 'Manny Perez': defaultdict(<class 'int'>, {'Fernando
Baez Mella': 1}), 'Jennifer Hale': defaultdict(<class 'int'>, {'Warren Sheppard':
1}), 'Mary Kate Wiles': defaultdict(<class 'int'>, {'Justin Paul Miller': 1}), 'C
arrie Bradstreet': defaultdict(<class 'int'>, {'Joel Paul Reisig': 1}), 'Elizabet
h Streb': defaultdict(<class 'int'>, {'Catherine Gund': 1}), 'Mike Stanley': defa
ulldict(<class 'int'>, {'Luke Dye': 1}), 'Patrick Kilpatrick': defaultdict(<class
'int'>, {'Michael Herz': 1}), 'Lawrence Gilliard Jr.': defaultdict(<class 'int'>,

```

```
{'Matty Rich': 1}}, 'Luis Sanchez': defaultdict(<class 'int'>, {'Alec Asten': 1}), 'John Karyus': defaultdict(<class 'int'>, {'Lloyd Kaufman': 1}), 'Jennifer Lafleur': defaultdict(<class 'int'>, {'Matthew Watts': 1}), 'Bob Proctor': defaultdict(<class 'int'>, {'Joe Kenemore': 1}), 'Chris Eigeman': defaultdict(<class 'int'>, {'Whit Stillman': 1}), 'Jack E. Currenton': defaultdict(<class 'int'>, {'Eric England': 1}), 'Micah Sloat': defaultdict(<class 'int'>, {'Oren Peli': 1}), 'Tony Alva': defaultdict(<class 'int'>, {'Stacy Peralta': 1}), 'Emily Rios': defaultdict(<class 'int'>, {'Richard Glatzer': 1}), 'Tony Doupe': defaultdict(<class 'int'>, {'Sue Corcoran': 1}), 'Greg Ayres': defaultdict(<class 'int'>, {'Jonathan Caouette': 1}), 'Bill Farmer': defaultdict(<class 'int'>, {'Ray Griggs': 1}), 'Catriona MacColl': defaultdict(<class 'int'>, {'Lucio Fulci': 1}), 'Brendon Small': defaultdict(<class 'int'>, {'': 1}), 'Treat Williams': defaultdict(<class 'int'>, {'Tom Vaughan': 1}), 'Dov Tiefenbach': defaultdict(<class 'int'>, {'Paul Fox': 1}), 'R. Brandon Johnson': defaultdict(<class 'int'>, {'Ari Kirschenbaum': 1, 'Steven Mena': 1}), 'Liv Ullmann': defaultdict(<class 'int'>, {'Ingmar Bergman': 1, 'Laslo Benedek': 1}), 'Lillian Gish': defaultdict(<class 'int'>, {'D.W. Griffith': 1}), 'Walter Koenig': defaultdict(<class 'int'>, {'Roger Nygard': 1}), 'Anita Page': defaultdict(<class 'int'>, {'Harry Beaumont': 1}), 'America Olivo': defaultdict(<class 'int'>, {'Franck Khalfoun': 1}), 'Amos Oz': defaultdict(<class 'int'>, {'Mor Loushy': 1}), 'Mark Zupan': defaultdict(<class 'int'>, {'Henry Alex Rubin': 1}), 'Michael Dudikoff': defaultdict(<class 'int'>, {'Sam Firstenberg': 1}), 'Carol Block': defaultdict(<class 'int'>, {'Doug Block': 1}), 'Tom E. Nicholson': defaultdict(<class 'int'>, {'Chad Kapper': 1}), 'Jack Warden': defaultdict(<class 'int'>, {'Sidney Lumet': 1}), 'Claudette Colbert': defaultdict(<class 'int'>, {'Frank Capra': 1}), 'Angeliki Papoulia': defaultdict(<class 'int'>, {'Yorgos Lanthimos': 1}), 'James Cagney': defaultdict(<class 'int'>, {'Lauren Lazin': 1}), 'Kimberly J. Brown': defaultdict(<class 'int'>, {'Gavin O'Connor': 1}), 'Simon Abkarian': defaultdict(<class 'int'>, {'Cédric Klapisch': 1}), 'Kevin Howarth': defaultdict(<class 'int'>, {'Niall Johnson': 1}), 'Daniel London': defaultdict(<class 'int'>, {'Kelly Reichardt': 1}), 'John Robinson': defaultdict(<class 'int'>, {'Kelly Reichardt': 1}), 'Embeth Davidtz': defaultdict(<class 'int'>, {'Eric Mendelsohn': 1}), 'Jean-Paul Belmondo': defaultdict(<class 'int'>, {'Jean-Luc Godard': 1}), 'Vera Ngassa': defaultdict(<class 'int'>, {'Florence Ayisi': 1}), 'Robert John Burke': defaultdict(<class 'int'>, {'Jesse Peretz': 1}), 'Don Wallace': defaultdict(<class 'int'>, {'Eddie O'Flaherty': 1}), 'Lalaine': defaultdict(<class 'int'>, {'Babar Ahmed': 1}), 'Trish Basinger': defaultdict(<class 'int'>, {'John D. Hancock': 1}), 'Tatyana Ali': defaultdict(<class 'int'>, {'Bruce Dellis': 1}), 'Michael Des Barres': defaultdict(<class 'int'>, {'Pece Dingo': 1}), 'Rumi Missabu': defaultdict(<class 'int'>, {'Étienne Faure': 1}), 'Dreema Walker': defaultdict(<class 'int'>, {'Craig Zobel': 1}), 'Aunjanue Ellis': defaultdict(<class 'int'>, {'Nicole Holofcener': 1}), 'Hector Echavarria': defaultdict(<class 'int'>, {'Ken Del Conte': 1}), 'Parry Shen': defaultdict(<class 'int'>, {'Justin Lin': 1}), 'Nicole Ari Parker': defaultdict(<class 'int'>, {'Maria Maggenti': 1}), 'Purva Bedi': defaultdict(<class 'int'>, {'Piyush Dinker Pandya': 1}), 'Keri Maletto': defaultdict(<class 'int'>, {'Daniel Cumbie': 1}), 'Charis Michelsen': defaultdict(<class 'int'>, {'Bill Plympton': 1}), 'Amir Talai': defaultdict(<class 'int'>, {'Greg Harrison': 1}), 'Marshal Hilton': defaultdict(<class 'int'>, {'Al Franklin': 1}), 'Ally Sheedy': defaultdict(<class 'int'>, {'Allison Anders': 1}), 'Samantha Futerman': defaultdict(<class 'int'>, {'Michael Kang': 1}), 'Nolan Gerard Funk': defaultdict(<class 'int'>, {'Paul Schrader': 1}), 'Judy Marte': defaultdict(<class 'int'>, {'Lori Silverbush': 1}), 'Michael Abbott Jr.': defaultdict(<class 'int'>, {'Jeff Nichols': 1}), 'James Urbaniak': defaultdict(<class 'int'>, {'Hilary Brougher': 1}), 'Dustin Fitzsimons': defaultdict(<class 'int'>, {'Rachel Goldenberg': 1}), 'Eme Ikwaakor': defaultdict(<class 'int'>, {'Jamin Winans': 1}), 'Florence Henderson': defaultdict(<class 'int'>, {'Tom Seidman': 1}), 'Victoria Jackson': defaultdict(<class 'int'>, {'Jason Naumann': 1}), 'Sam Adams': defaultdict(<class 'int'>, {'Sam Martin': 1}), 'Noah Segal': defaultdict(<class 'int'>, {'Andrew Hyatt': 1}), 'Joshua Ray Bell': defaultdict(<class 'int'>, {'Joel Paul Reising': 1}), 'John Gilbert': defaultdict(<class 'int'>, {'King Vidor': 1}), 'Matthew Emerick': defaultdict(<class 'int'>, {'Zoran Lisinac': 1}), 'Mark Fantas
```

```

ia': defaultdict(<class 'int'>, {"James O'Brien": 1}), 'Brendan Doogie Milewski':
defaultdict(<class 'int'>, {'Tom Putnam': 1}), 'Dan Futterman': defaultdict(<clas
s 'int'>, {'Jon Shear': 1}), 'Christina Hart': defaultdict(<class 'int'>, {'Al Si
lliman Jr.': 1}), 'Kenneth Tobey': defaultdict(<class 'int'>, {'Eugène Lourié':
1}), 'Hugh Keays-Byrne': defaultdict(<class 'int'>, {'George Miller': 1}), 'Mae W
est': defaultdict(<class 'int'>, {'Lowell Sherman': 1}), 'The Dalai Lama': defaul
tdict(<class 'int'>, {'Maurizio Benazzo': 1}), 'Michael Joiner': defaultdict(<cla
ss 'int'>, {'David G. Evans': 1}), 'Devid Striesow': defaultdict(<class 'int'>,
{'Tom Tykwer': 1}), 'William Joseph Elk III': defaultdict(<class 'int'>, {'Sherma
n Alexie': 1}), 'Matisyahu': defaultdict(<class 'int'>, {'Justin Dillon': 1}), 'M
arri Savinar': defaultdict(<class 'int'>, {'Adam Rifkin': 1}), 'Melanie Papalia':
defaultdict(<class 'int'>, {'Eric Valette': 1}), 'Susan Chambers': defaultdict(<c
lass 'int'>, {'Livingston Oden': 1}), 'Marina Vlady': defaultdict(<class 'int'>,
{'Chris Marker': 1}), 'Hanne Aagesen': defaultdict(<class 'int'>, {'Carl Theodor
Dreyer': 1}), 'William Katt': defaultdict(<class 'int'>, {'Richard Schenkman':
1}), 'Darryl Hunt': defaultdict(<class 'int'>, {'Ricki Stern': 1}), 'John Newto
n': defaultdict(<class 'int'>, {'James Kerwin': 1}), 'Elaine Bromka': defaultdict
(<class 'int'>, {'C. Fraser Press': 1}), 'Paul Hickert': defaultdict(<class 'in
t'>, {'Rania Attieh': 1}), 'Bahare Seddiqi': defaultdict(<class 'int'>, {'Majid M
ajidi': 1}), 'S. Epatha Merkerson': defaultdict(<class 'int'>, {'Spike Lee': 1}),
'Abigail Evans': defaultdict(<class 'int'>, {'Cary Bell': 1}), 'Iulia Ciochina':
defaultdict(<class 'int'>, {'Nicolae Constantin Tanase': 1}), 'Robin Lord Taylo
r': defaultdict(<class 'int'>, {'Mike Cahill': 1}), 'Charla Cochran': defaultdict
(<class 'int'>, {'Ken Roht': 1}), 'Bebe Neuwirth': defaultdict(<class 'int'>, {'G
ary Winick': 1}), 'Glen Hansard': defaultdict(<class 'int'>, {'John Carney': 1}),
'Marilyn Rising': defaultdict(<class 'int'>, {'Robinson Devor': 1}), 'Temple Gran
din': defaultdict(<class 'int'>, {'Michel Orion Scott': 1}), 'Alan Brent': defaul
tdict(<class 'int'>, {'Pat Holden': 1}), 'Joel Moody': defaultdict(<class 'int'>,
{'Eric Bugbee': 1}), 'Peter Robbins': defaultdict(<class 'int'>, {'Bill Melende
z': 1}), 'Naderev Sano': defaultdict(<class 'int'>, {'Dena Seidel': 1}), 'Lisa An
n': defaultdict(<class 'int'>, {'Deborah Anderson': 1}), 'Ariel Hsing': defaulttdi
ct(<class 'int'>, {'Sara Newens': 1}), 'J.D. Williams': defaultdict(<class 'in
t'>, {'Sai Varadan': 1}), 'Christopher Denham': defaultdict(<class 'int'>, {'Zal
Batmanglij': 1}), 'Mark Duplass': defaultdict(<class 'int'>, {'Lynn Shelton': 1,
'Jay Duplass': 1}), 'Diane Sorrentino': defaultdict(<class 'int'>, {'Valentine':
1}), 'Philippe Leroy': defaultdict(<class 'int'>, {'Jean-Luc Godard': 1}), 'Jason
Stuart': defaultdict(<class 'int'>, {'Nate Parker': 1}), 'Richard Moll': defaulttd
ict(<class 'int'>, {'Nathan Smith Jones': 1}), 'Pfeifer Brown': defaultdict(<clas
s 'int'>, {'Travis Cluff': 1}), 'Hal Landon Jr.': defaultdict(<class 'int'>, {'Da
vid Lynch': 1}), 'Stephen Carr': defaultdict(<class 'int'>, {'Harry F. Millarde':
1}), 'Robert Townsend': defaultdict(<class 'int'>, {'Robert Townsend': 1}), 'Leon
Isaac Kennedy': defaultdict(<class 'int'>, {'Jamaa Fanaka': 1}), 'Fay Masterson':
defaultdict(<class 'int'>, {'Larry Blamire': 1}), 'Kevin P. Farley': defaultdict
(<class 'int'>, {'Stephen Langford': 1}), 'Jonathan Blow': defaultdict(<class 'in
t'>, {'Lisanne Pajot': 1}), 'Sheldon Leonard': defaultdict(<class 'int'>, {'John
Reinhardt': 1}), 'Claire Gordon-Harper': defaultdict(<class 'int'>, {'Patrick Rya
n Sims': 1}), 'Jon Brion': defaultdict(<class 'int'>, {'Jason Miller': 1}), 'Jack
McGee': defaultdict(<class 'int'>, {'Julie Davis': 1}), 'Michelle Simone Miller':
defaultdict(<class 'int'>, {'Brett Piper': 1}), 'Chemeeeka Walker': defaultdict(<c
lass 'int'>, {'Morgan Spurlock': 1}), 'Virginia Leith': defaultdict(<class 'in
t'>, {'Joseph Green': 1}), 'Ty Parker': defaultdict(<class 'int'>, {'Wade Gasqu
e': 1}), 'Justin Gordon': defaultdict(<class 'int'>, {'Mike Flanagan': 1}), 'Shar
i Albert': defaultdict(<class 'int'>, {'Edward Burns': 1}), 'Paul Daniel Ayotte':
defaultdict(<class 'int'>, {'Matt Johnson': 1}), 'Marcello Mastroianni': defaulttd
ict(<class 'int'>, {'Bruno Barreto': 1}), 'Dan Novy': defaultdict(<class 'int'>,
{'Andrew Leman': 1}), "Chris 'Wonder' Schoeck": defaultdict(<class 'int'>, {'Dave
Carroll': 1}), 'Taliesin Jaffe': defaultdict(<class 'int'>, {'Patrick Meaney':
1}), 'Jan Haley': defaultdict(<class 'int'>, {'Chad Hartigan': 1}), 'Paul Schneid
er': defaultdict(<class 'int'>, {'David Gordon Green': 1}), 'Derick Martini': def
aultdict(<class 'int'>, {'Kevin Jordan': 1}), 'Elissa Dowling': defaultdict(<clas

```

```
s 'int'>, {'Travis Legge': 1}), 'Ringo Starr': defaultdict(<class 'int'>, {'Martin Scorsese': 1}), 'Julianne Gabert': defaultdict(<class 'int'>, {'Collin Joseph Neal': 1}), 'Joseph Campanella': defaultdict(<class 'int'>, {'Mike Bruce': 1}), 'Andrew Bujalski': defaultdict(<class 'int'>, {'Andrew Bujalski': 2}), 'Nichole Ceballos': defaultdict(<class 'int'>, {'Damir Catic': 1}), 'Don Brooks': defaultdict(<class 'int'>, {'James Bidgood': 1}), 'Stacy Edwards': defaultdict(<class 'int'>, {'Neil LaBute': 1}), 'Franky G': defaultdict(<class 'int'>, {'Eric Eason': 1}), 'Tommy Pallotta': defaultdict(<class 'int'>, {'Richard Linklater': 1}), 'Tjasa Ferme': defaultdict(<class 'int'>, {'Joseph Mazzella': 1}), 'Kristen Seavey': defaultdict(<class 'int'>, {'Travis Legge': 1}), 'Shannen Fields': defaultdict(<class 'int'>, {'Alex Kendrick': 1}), 'Ashley Tramonte': defaultdict(<class 'int'>, {'Marcus Nispel': 1}), 'Robbie Barnes': defaultdict(<class 'int'>, {'Brandon Landers': 1}), 'Paul Ogola': defaultdict(<class 'int'>, {'Jim Chuchu': 1}), 'Zoe Lister-Jones': defaultdict(<class 'int'>, {'Daryl Wein': 1}), 'Sean Whalen': defaultdict(<class 'int'>, {'Jason Trost': 1}), 'Divine': defaultdict(<class 'int'>, {'John Waters': 1}), 'Maggie Cheung': defaultdict(<class 'int'>, {'Olivier Assayas': 1}), 'Fereshteh Sadre Orafaiy': defaultdict(<class 'int'>, {'Jafar Panahi': 1}), 'Patrick O'Donnell': defaultdict(<class 'int'>, {'Ivan Kavanagh': 1}), 'Kôji Yakucho': defaultdict(<class 'int'>, {'Kiyoshi Kurosawa': 1}), 'Tatiana Suarez-Pico': defaultdict(<class 'int'>, {'Tadeo Garcia': 1}), 'Julianna Pitt': defaultdict(<class 'int'>, {'Thomas L. Phillips': 1}), 'Shane Carruth': defaultdict(<class 'int'>, {'Shane Carruth': 1}), 'Ian Gamazon': defaultdict(<class 'int'>, {'Neill Dela Llana': 1}), 'Carlos Gallardo': defaultdict(<class 'int'>, {'Robert Rodriguez': 1}), 'Richard Jewell': defaultdict(<class 'int'>, {'Anthony Vallone': 1}), 'Kerry Bishé': defaultdict(<class 'int'>, {'Edward Burns': 1}), 'Natalie Zea': defaultdict(<class 'int'>, {'': 1}), 'Eva Boehnke': defaultdict(<class 'int'>, {'Benjamin Roberds': 1}), 'John August': defaultdict(<class 'int'>, {'Jon Gunn': 1})})
```

B.4. Print seleccionado

Demasiada información. Deseamos imprimir únicamente, para cada actor, las colaboraciones que superen un mínimo, dato entrada. Si un actor no tiene ninguna colaboración que supere dicho mínimo, lógicamente no debe mostrarse.

In [96]: *# Esta celda debe ser completada por el estudiante*

```
def print_collaborations_min(
    conteo_colaboraciones: defaultdict, colaboraciones_minimas: int
) -> None:
    """
    Recibe un diccionario, cuya clave es una tupla (título, año), y un \
    número entero, que es el número mínimo de colaboraciones a considerar. \
    La función no devuelve nada, pues simplemente imprime el nombre de cada \
    actor, y una lista formada por n tuplas, con el nombre del director y el \
    número de colaboraciones de ese actor con dicho director.

    Parameters
    -----
    conteo_colaboraciones : Un diccionario con las colaboraciones de cada \
    actor con cada director y el recuento de las mismas.
    colaboraciones_minimas : El número mínimo de colaboraciones que tiene \
    que tener el actor con el director para que se muestren.

    Returns
    -----
    None
    """
```

```
colaboraciones_por_actor = defaultdict(list)

for director, actores in conteo_colaboraciones.items():
    for actor, colabs in actores.items():
        if colabs >= colaboraciones_minimas:
            colaboraciones_por_actor[actor].append((director, colabs))

for actor, apariciones in colaboraciones_por_actor.items():
    print(f"{actor} -> {apariciones}")
```

In [97]: *# Test de funcionamiento*

```
print_collaborations_min(num_collaborations, 5)
```

```
Gore Verbinski -> [('Johnny Depp', 5)]
Tim Burton -> [('Johnny Depp', 6)]
Martin Scorsese -> [('Leonardo DiCaprio', 5), ('Robert De Niro', 7)]
Wes Anderson -> [('Bill Murray', 5)]
Clint Eastwood -> [('Clint Eastwood', 10)]
Woody Allen -> [('Woody Allen', 10)]
```

C. Algunos gráficos sencillos [1 punto]

C.1 Un modelo de gráfica

Vamos a diseñar un modelo de gráfica sencillo que nos sirva para las siguientes representaciones. Tomará como parámetro una lista de pares (x, y) , y opcionalmente los tres rótulos explicativos que necesitamos incluir. Además, queremos que las etiquetas de las abscisas aparezcan inclinadas, para poder luego mostrar intervalos de edad.

Las pruebas de funcionamiento te darán más información que las explicaciones que pueda yo dar aquí.

In [98]: *# Esta celda debe ser completada por el estudiante*

```
def representar_xxx_yyy(
    lista_puntos: list[tuple[float, float]], titulos=None
) -> None:
    """
    Recibe una lista de puntos (tuplas de float), y un parámetro opcional, \
    títulos, que contendrá, por orden, el título del gráfico, la etiqueta \
    del eje X y la etiqueta del eje Y, en caso de que se desee que se \
    muestren. La función mostrará un gráfico con dichos puntos.

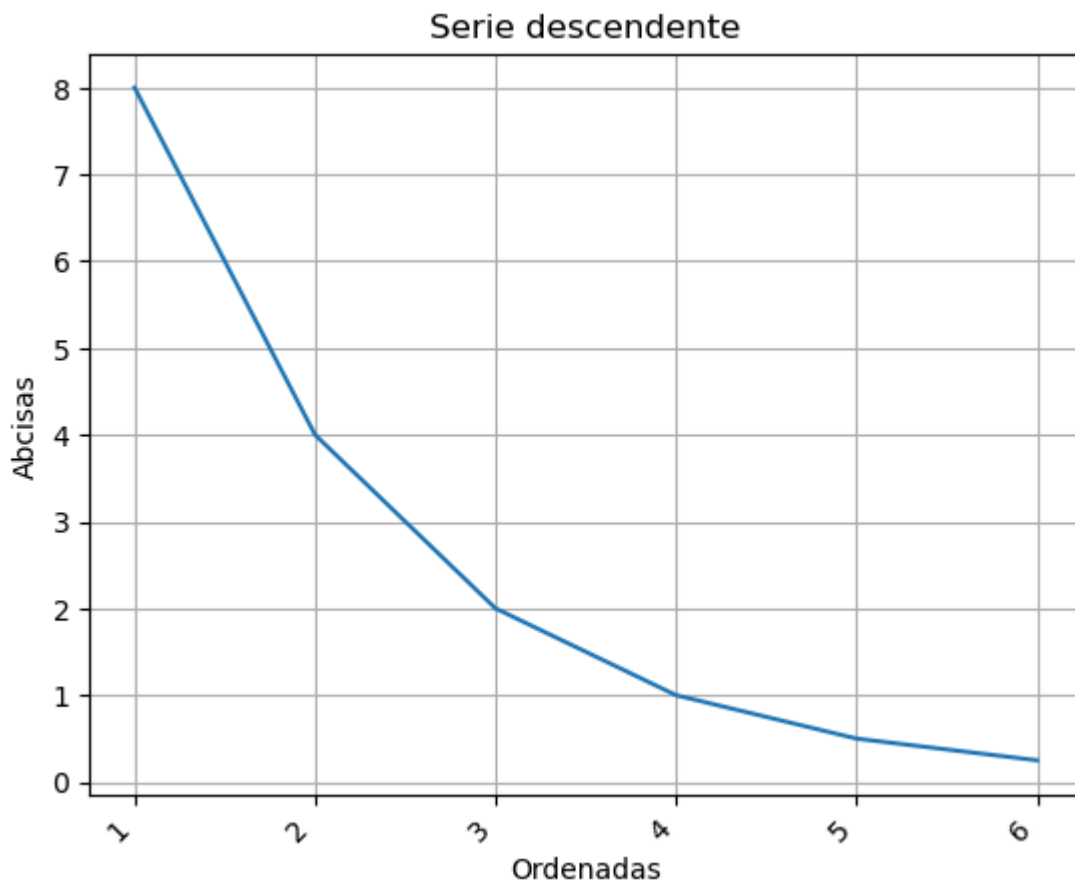
    Parameters
    -----
    lista_puntos : Una lista con pares (x, y) de puntos
    títulos : Contendrá los parámetros **título del gráfico**, **label del\
    eje X** y **label del eje Y**. Puede contener uno, dos, los tres o \
    ninguno. Por defecto, es igual a None.

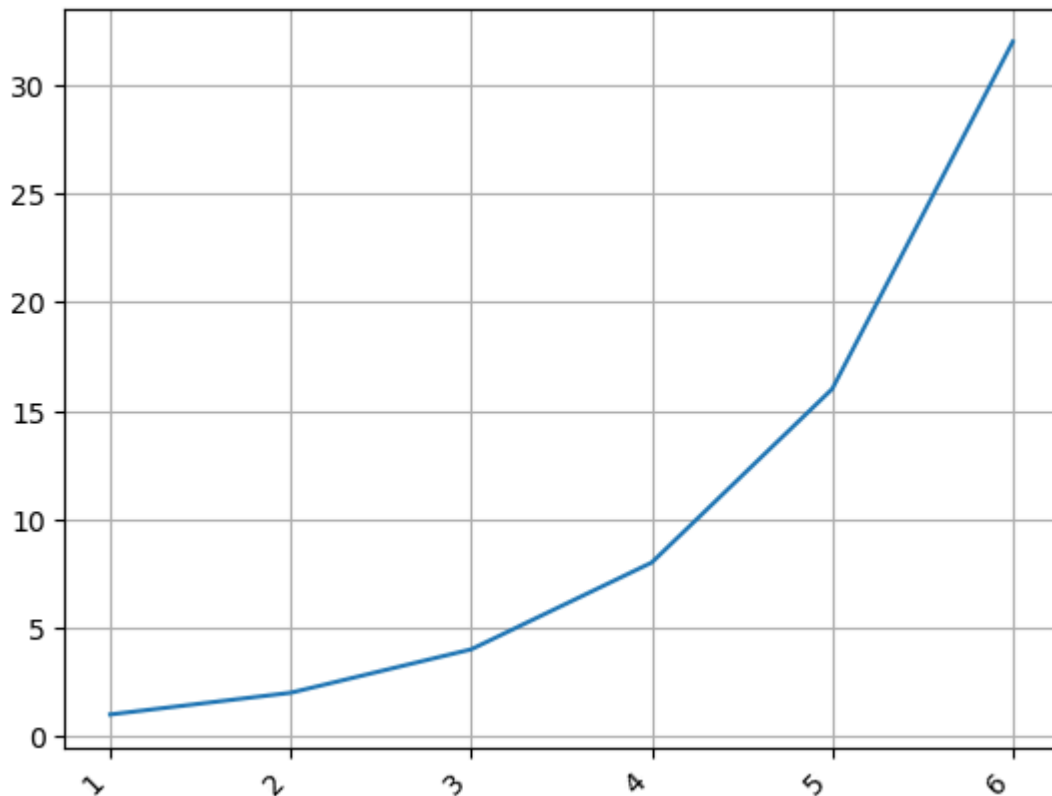
    Returns
    -----
    None
    """
```

```
# Descomponemos lista_puntos en dos listas, una para los valores de x
# y otra para los valores de y
x_vals, y_vals = zip(*lista_puntos)
plt.plot(x_vals, y_vals) # Graficamos los puntos
plt.grid(True) # Mostramos la cuadrícula
# Comprobamos si se ha pasado el parámetro de títulos y, en caso de que
# se haya pasado, cuál es su longitud para ver que argumentos aplican
if titulos:
    plt.title(titulos[0]) # Mostramos el título
    if len(titulos) == 2: # Si hay 2 parámetros, ponemos label al eje X
        plt.xlabel(titulos[1])
    if len(titulos) == 3: # Si hay 3 parámetros, ponemos label a X e Y
        plt.xlabel(titulos[1])
        plt.ylabel(titulos[2])
plt.xticks(rotation=45, ha='right') # Para rotar los labels del eje X
plt.show() # Mostramos el gráfico
```

In [99]: # Pruebas de funcionamiento:

```
representar_xxx_yyy([(1, 8), (2, 4), (3, 2), (4, 1), (5, 0.5), (6, 0.25)], ["Ser
representar_xxx_yyy([(1, 1), (2, 2), (3, 4), (4, 8), (5, 16), (6, 32)])
```





Una gráfica concreta

Deseamos representar el número de películas de nuestra base de datos que se han producido en un intervalo de años dado.

In [100...

Esta celda debe ser completada por el estudiante

```
def repr_movies_years(
    diccionario_peliculas: dict[tuple[str, int], list[str]],
    inicio: int,
    fin: int
) -> None:
    """
    Recibe un diccionario con todas las películas en una base de datos, cuya \
    clave es una tupla (título, año) y sus valores una lista de strings con \
    los diferentes campos; y dos enteros, que representan el inicio y el fin \
    del intervalo de los datos a representar. Dibuja un gráfico para dicho \
    intervalo, con el número de películas producidas en cada año.

    Parameters
    -----
        diccionario_peliculas : Un diccionario con todas las películas.
        inicio : El inicio del intervalo de años a representar.
        fin : El fin del intervalo de años (debe ser mayor que inicio).

    Returns
    -----
        None
    """

    # Obtenemos el número de películas por año con la función que creamos
    datos = years_num_movies(diccionario_peliculas, inicio, fin)
    # La función devuelve un diccionario: sus claves son los años;
```

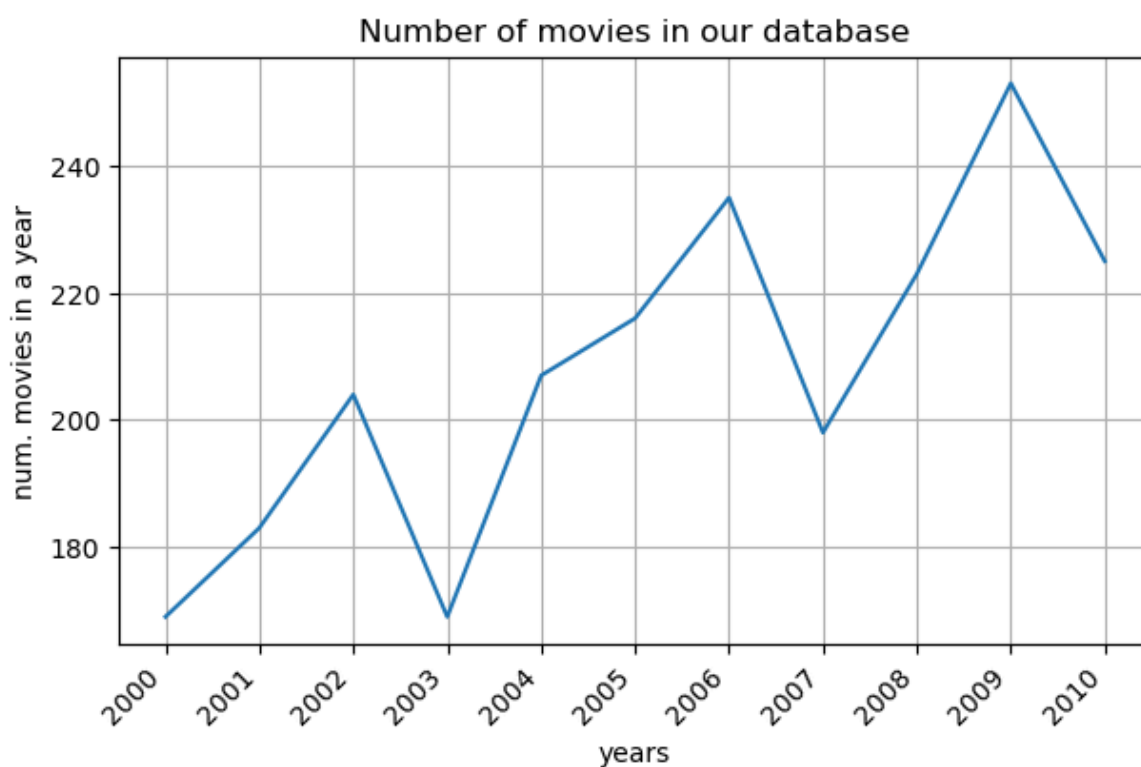
```
# sus valores, el recuento de películas
anhos = list(datos.keys())
conteo = list(datos.values())

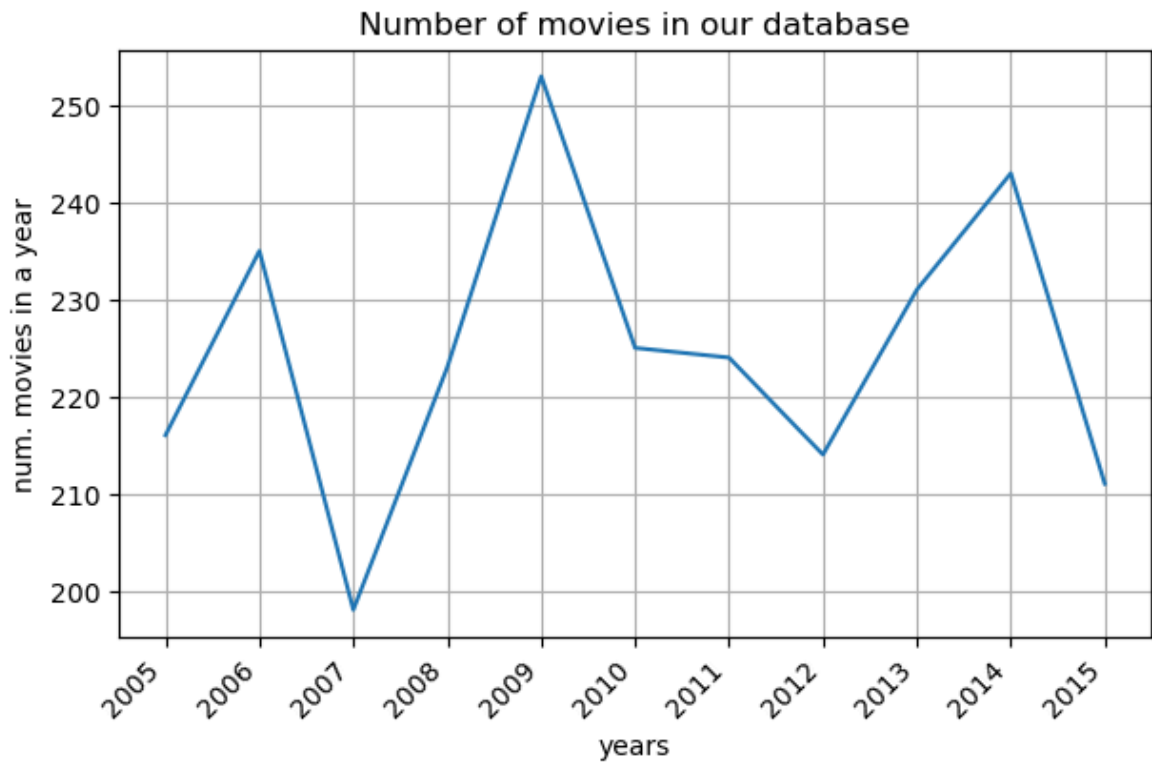
plt.figure(figsize=(7,4))
plt.plot(anhos, conteo)
plt.grid(True)
plt.title("Number of movies in our database")
plt.xlabel("years")
plt.ylabel("num. movies in a year")
plt.xticks(anhos, rotation=45, ha='right')
plt.show()
```

In [101...

Test de funcionamiento

```
repr_movies_years(main_dict_data, 2000, 2010)
repr_movies_years(main_dict_data, 2005, 2015)
```





D. Acceso a las urls de imdb y webscraping [2 puntos]

D.1. Recuperación de las URLs

Con sencillas instrucciones, deseamos recuperar todas las *urls* de las películas de nuestro archivo, mostrando cuántas son, la primera de ellas o las diez primeras por ejemplo:

```
In [102... # Esta celda debe ser completada por el estudiante

# El diccionario tiene la url en los valores, en la columna octava.
# De esta forma, obtenemos todas las urls en una lista
urls = [url[1][7] for url in main_dict_data.items()]
first_url_movie = urls[0] # La primera url es la 0
first_ten_urls = urls[:10] # Para las 10 primeras, usaremos un slice
```

```
In [103... # Test de funcionamiento

print(len(urls))

print()

print(first_url_movie)

print()

print(first_ten_urls)
```

4919

<http://www.imdb.com/title/tt0499549/>

```
['http://www.imdb.com/title/tt0499549/', 'http://www.imdb.com/title/tt0449088/',
'http://www.imdb.com/title/tt2379713/', 'http://www.imdb.com/title/tt1345836/',
'http://www.imdb.com/title/tt5289954/', 'http://www.imdb.com/title/tt0401729/',
'http://www.imdb.com/title/tt0413300/', 'http://www.imdb.com/title/tt0398286/',
'http://www.imdb.com/title/tt2395427/', 'http://www.imdb.com/title/tt0417741/']
```

In [104...

```
# Una forma de evitar el error "Requests 403 forbidden" al hacer web scraping en

HEADERS = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/a
    "Accept-Language": "en-US,en;q=0.9",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "keep-alive",
    "Upgrade-Insecure-Requests": "1",
    "Sec-Fetch-Dest": "document",
    "Sec-Fetch-Mode": "navigate",
    "Sec-Fetch-Site": "none",
    "Sec-Fetch-User": "?1",
    "Cache-Control": "max-age=0",
}
```

D.2. Carga de la estructura sintáctica de una URL

Ahora, deseamos extraer el código `html` de una película.

In [105...

```
# Esta celda debe ser completada por el estudiante

def soup_movie(url: str) -> BeautifulSoup :
    """
    Recibe una url (en un string) y realiza una solicitud HTTP a dicha url \
    devolviendo su contenido parseado en un objeto BeautifulSoup.

    Parameters
    -----
        url : Un string que representa una url.

    Returns
    -----
        BeautifulSoup : El contenido HTML de la página.
    """

    # Hace un get a la url, con los encabezados HEADERS, y obtiene su contenido
    # mediante el .text, indicando que use el parser HTML de python
    return BeautifulSoup(requests.get(url, headers=HEADERS).text,
                          'html.parser')
```

In [106...

```
# Test de funcionamiento

soup = soup_movie(first_url_movie)

print(str(soup)[:1000])
print()
print("... ..")
```

```
print()
print(str(soup)[-1000:])
```

```
<!DOCTYPE html>
<html lang="en-US" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://
opengraphprotocol.org/schema/"><head><meta charset="utf-8"/><meta content="width=
device-width" name="viewport"/><script>if(typeof uet === 'function'){ uet('bb',
'LoadTitle', {wb: 1}); }</script><script>window.addEventListener('load', (event)
=> {
    if (typeof window.csa !== 'undefined' && typeof window.csa === 'functio
n') {
        var csaLatencyPlugin = window.csa('Content', {
            element: {
                slotId: 'LoadTitle',
                type: 'service-call'
            }
        });
        csaLatencyPlugin('mark', 'clickToBodyBegin', 1731615259331);
    }
})</script><title>Avatar (2009) - IMDb</title><meta content="Avatar: Directed
by James Cameron. With Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen La
ng. A paraplegic Marine dispatched to the moon Pandora on a unique mission become
s torn between following his orders and pr

... ..
... ..

g(e+c);return!!e}function n(){for(var e=RegExp("^https://(.*\\.(images|ssl-images|
media)-amazon\\.com|"+c.location.hostname+)/images/", "i"), d={}, h=0, k=c.performanc
e.getEntriesByType("resource"), l=!1, b, a, m, f=0; f<k.length; f++)if(a=k[f], 0<a.transf
erSize&&a.transferSize>=a.encodedBodySize&&(b=e.exec(String(a.name)))&&3===b.leng
th){a:{b=a.serverTiming||[];for(a=0;a<b.length;a++)if("provider"===b[a].name){b=b
[a].description;break a}b=void 0}b&&(1||(1=g(b, "_cdn_fr")),
a=d[b]=(d[b]||0)+1,a>h&&(m=b, h=a))}g(m, "_cdn_mp")}d.ue&&"function"===typeof d.ue.
tag&&c.performance&&c.location&&n()), "cdnTagging")(ue_csm, window);

}

/* △ */
</script>
</div>
<noscript>

</noscript>
<script>window.ue && ue.count && ue.count('CSMLibrarySize', 61498)</script></div>
</body></html>
```

D.3. Extracción de algunas piezas de información de una URL

Y ahora, con dicho código, deseamos extraer la siguiente información, referida a la película `first_url_movie`:

- La etiqueta completa del título de la película
- La descripción (sólo el contenido).

- La lista de los actores del *reparto principal*
- La información sobre el presupuesto

In [107...

```
# Esta celda debe ser completada por el estudiante

def gatherActorsFromMovie(soup: BeautifulSoup) -> list[str]:
    """
    Recibe un objeto de tipo BeautifulSoup con el contenido HTML de una url \
    de una película y devuelve una lista con todos los actores de la misma.

    Parameters
    -----
    soup : Un objeto BeautifulSoup con el contenido HTML de una url.

    Returns
    -----
    list[str] : Una lista con todos los actores de la película.
    """

    # Los actores están en un elemento <a>, cuyo atributo 'data-testid' toma el
    # valor 'title-cast-item__actor' (se puede comprobar visualmente en la url)
    rep = soup.find_all('a', attrs={'data-testid': 'title-cast-item__actor'})

    # Devolvemos el contenido del texto, en una lista
    return [actor.text for actor in rep]

# Mostramos la etiqueta del título
print(soup.find('title'))
print()
# Mostramos el contenido de la descripción de la película, que está en un
# elemento <meta>, donde el atributo "name" es "description"
print(soup.find('meta', attrs={'name': 'description'})['content'])
print()
# Usamos la función definida anteriormente para mostrar los actores
print(gatherActorsFromMovie(soup))
print()
# Para el presupuesto, tenemos que encontrar el elemento <span>, y buscar el
# atributo class que cuyo valor es 'ipc-metadata-list-item__list-content-item',
# estando el presupuesto en el segundo elemento
print(soup.find_all(
    'span', attrs={'class': 'ipc-metadata-list-item__list-content-item'}
)[2].text)
```

<title>Avatar (2009) - IMDb</title>

Avatar: Directed by James Cameron. With Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang. A paraplegic Marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.

['Sam Worthington', 'Zoe Saldana', 'Sigourney Weaver', 'Michelle Rodriguez', 'Stephen Lang', 'Giovanni Ribisi', 'Joel David Moore', 'CCH Pounder', 'Wes Studi', 'Laz Alonso', 'Dileep Rao', 'Matt Gerald', 'Sean Anthony Moran', 'Jason Whyte', 'Scott Lawrence', 'Kelly Kilgour', 'James Patrick Pitt', 'Sean Patrick Murphy']

\$237,000,000 (estimated)

D.4. Actores que intervienen en una lista de URLs

Necesitamos crear un archivo con los actores del reparto principal de las películas de IMDB, dada la lista de sus URLs.

In [108...

Esta celda debe ser completada por el estudiante

```
def gather_actors(archivo: str, urls: list[str]) -> None:
    """
```

Recibe un string, que representa la ruta a un archivo, y una lista de \
urls de películas, y crea un archivo con todos los actores del reparto \
principal de dichas películas.

Parameters

archivo : Una cadena de texto con la ruta al archivo donde se \
guardarán los datos

urls : Una lista con las urls de las películas.

Returns

None

"""

```
with open(archivo, 'w') as file:
```

```
    for url in urls: # Recorremos las urls
```

```
        # Hacemos la petición HTTP para acceder al contenido
```

```
        soup = BeautifulSoup(requests.get(url, headers=HEADERS).text,  
                              'html.parser')
```

```
        # Obtenemos la lista de los actores
```

```
        actores = gatherActorsFromMovie(soup)
```

```
        # Para cada 'actor' en la lista de actores, añadimos un salto de  
        # línea al final de cada nombre. De esta forma, cada actor saldrá  
        # en una línea, y nos permite escribir todos los actores obtenidos  
        # de una misma url a la vez al archivo
```

```
        actores = map(lambda actor: actor + '\n', actores)
```

```
        # Escribimos todos los actores al archivo
```

```
        file.writelines(actores)
```

In [109...

Test de funcionamiento

```
# OJO: esta operación puede llevar bastante tiempo.
```

```
# Para esta prueba, usamos un número limitado de películas.
```

```
gather_actors("actors_3_first_movies.txt", urls[:3])
```

```
! type actors_3_first_movies.txt
```

Sam Worthington
Zoe Saldana
Sigourney Weaver
Michelle Rodriguez
Stephen Lang
Giovanni Ribisi
Joel David Moore
CCH Pounder
Wes Studi
Laz Alonso
Dileep Rao
Matt Gerald
Sean Anthony Moran
Jason Whyte
Scott Lawrence
Kelly Kilgour
James Patrick Pitt
Sean Patrick Murphy
Johnny Depp
Orlando Bloom
Keira Knightley
Geoffrey Rush
Jack Davenport
Bill Nighy
Jonathan Pryce
Lee Arenberg
Mackenzie Crook
Kevin McNally
David Bailie
Stellan Skarsgård
Tom Hollander
Naomie Harris
Martin Klebba
David Schofield
Lauren Maher
Dermot Keaney
Daniel Craig
Christoph Waltz
Léa Seydoux
Ralph Fiennes
Monica Bellucci
Ben Whishaw
Naomie Harris
Dave Bautista
Andrew Scott
Rory Kinnear
Jesper Christensen
Alessandro Cremona
Stephanie Sigman
Tenoch Huerta
Adriana Paz
Domenico Fortunato
Marco Zingaro
Stefano Elfi DiClaudia

```
In [110... # La siguiente llamada llevaría un tiempo realmente largo:  
  
# import time # para cronometrar esta función, que tarda mucho  
  
# reloj_inicio = time.time()
```

```
# gather_actors("actors_all_movies.txt", urls)
# reloj_fin = time.time()

# print("Tiempo invertido: %s segundos." % (reloj_fin - reloj_inicio))
```

E. Pandas [2 puntos]

E.1. El primer paso es la carga del archivo en un dataframe

```
In [111... # Esta celda debe ser completada por el estudiante

def load_dataframe(archivo: str) -> pd.DataFrame:
    """
    Recibe un string, que representa la ruta a un archivo, y crea un \
    dataframe de pandas con todos los datos de dicho archivo.

    Parameters
    -----
        archivo : Ruta al archivo desde el que se cargaran los datos.

    Returns
    -----
        pd.DataFrame : Un dataframe en el que se han cargado todos los datos \
        del archivo
    """

    return pd.read_csv(archivo)
```

```
In [112... # Test de funcionamiento

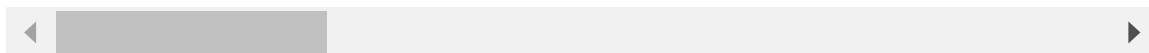
tabla_completa = load_dataframe(MOVIES_DATA)

tabla_completa
```

Out[112...

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	ac
0	Color	James Cameron	723.0	178.0	0.0	
1	Color	Gore Verbinski	302.0	169.0	563.0	
2	Color	Sam Mendes	602.0	148.0	0.0	
3	Color	Christopher Nolan	813.0	164.0	22000.0	
4	NaN	Doug Walker	NaN	NaN	131.0	
...	
5038	Color	Scott Smith	1.0	87.0	2.0	
5039	Color	NaN	43.0	43.0	NaN	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	
5042	Color	Jon Gunn	43.0	90.0	16.0	

5043 rows × 28 columns



E.2. Tabla de los campos principales

A partir de la tabla anterior, construimos otra con sólo algunos de los campos:

In [113...

```
# Esta celda debe ser completada por el estudiante

def fields_selected_dataframe(data_datos: pd.DataFrame) -> pd.DataFrame:
    """
    Recibe un dataframe de pandas y, a partir de una variable global llamada \
    "CAMPOS_REDUCIDOS", devuelve otro dataframe únicamente con esos campos.

    Parameters
    -----
    data_datos : El dataframe completo, con todos los datos.

    Returns
    -----
    pd.DataFrame : Un sub-dataframe con los campos de interés.
    """

    return data_datos[CAMPOS_REDUCIDOS]
```

In [114...

```
# test de comprobación

tabla_breve = fields_selected_dataframe(tabla_completa)
```


tabla_breve

Out[114...

	movie_title	color	director_name	language	country	actor_1_name	
0	Avatar	Color	James Cameron	English	USA	CCH Pounder	http://www.ir
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	English	USA	Johnny Depp	http://www.ir
2	Spectre	Color	Sam Mendes	English	UK	Christoph Waltz	http://www.ir
3	The Dark Knight Rises	Color	Christopher Nolan	English	USA	Tom Hardy	http://www.ir
4	Star Wars: Episode VII - The Force Awakens ...	NaN	Doug Walker	NaN	NaN	Doug Walker	http://www.ir
...
5038	Signed Sealed Delivered	Color	Scott Smith	English	Canada	Eric Mabius	http://www.ir
5039	The Following	Color	NaN	English	USA	Natalie Zea	http://www.ir
5040	A Plague So Pleasant	Color	Benjamin Roberds	English	USA	Eva Boehnke	http://www.ir
5041	Shanghai Calling	Color	Daniel Hsia	English	USA	Alan Ruck	http://www.ir
5042	My Date with Drew	Color	Jon Gunn	English	USA	John August	http://www.ir

5043 rows × 7 columns



E.3. Columnas de una tabla

¿Cuáles son las columnas de nuestra `tabla_breve` ?

In [115...

```
# Esta celda debe ser completada por el estudiante

# Las columnas de un dataframe se pueden obtener mediante el atributo columns
tabla_breve.columns
```

Out[115...

```
Index(['movie_title', 'color', 'director_name', 'language', 'country',
      'actor_1_name', 'movie_imdb_link'],
      dtype='object')
```

E.4. Campos missing

Algunos campos muestran un valor `NaN`. Deseamos cambiarlo por una cadena de caracteres: `"Desc"`.

In [116...

```
# Esta celda debe ser completada por el estudiante

# Aplicamos la función fillna de los DataFrames para completar los valores
# NaN con la cadena "Desc"
tabla_breve = tabla_breve.fillna("Desc")
```

In [117...

```
# Test de comprobación
```

```
tabla_breve
```

Out[117...

	movie_title	color	director_name	language	country	actor_1_name	
0	Avatar	Color	James Cameron	English	USA	CCH Pounder	http://www.ir
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	English	USA	Johnny Depp	http://www.ir
2	Spectre	Color	Sam Mendes	English	UK	Christoph Waltz	http://www.ir
3	The Dark Knight Rises	Color	Christopher Nolan	English	USA	Tom Hardy	http://www.ir
4	Star Wars: Episode VII - The Force Awakens ...	Desc	Doug Walker	Desc	Desc	Doug Walker	http://www.ir
...
5038	Signed Sealed Delivered	Color	Scott Smith	English	Canada	Eric Mabius	http://www.ir
5039	The Following	Color	Desc	English	USA	Natalie Zea	http://www.ir
5040	A Plague So Pleasant	Color	Benjamin Roberds	English	USA	Eva Boehnke	http://www.ir
5041	Shanghai Calling	Color	Daniel Hsia	English	USA	Alan Ruck	http://www.ir
5042	My Date with Drew	Color	Jon Gunn	English	USA	John August	http://www.ir

5043 rows × 7 columns



E.5. Director → películas y número de películas

Función que averigua la lista de títulos de películas de un director dado:

In [118...

```
# Esta celda debe ser completada por el estudiante
```

```
def titulos_de_director_df(
    tab_df: pd.DataFrame,
    director: str
) -> pd.DataFrame:
    """
    Recibe un dataframe de pandas y el nombre de un director, y devuelve una \
    tabla solamente con los títulos que ha dirigido ese director.

    Parameters
    -----
        tab_df : El dataframe completo, con todos los datos.
        director : Un string, que representa el nombre del director.

    Returns
    -----
        pd.DataFrame : Una tabla con las películas dirigidas por 'director'.
    """

    # Devolvemos el 'movie_title' para aquellas filas en las que
    # 'director_name' sea el nombre buscado
    return tab_df[tab_df['director_name'] == director][['movie_title']]
```

In [119...

```
# Test de comprobación:

tabla_tits = titulos_de_director_df(tabla_breve, "James Cameron")
tabla_tits
```

Out[119...

	movie_title
0	Avatar
26	Titanic
288	Terminator 2: Judgment Day
291	True Lies
606	The Abyss
2486	Aliens
3575	The Terminator

In [120...

```
# Esta celda debe ser completada por el estudiante
```

```
list_tits = tabla_tits["movie_title"].to_list()
print(list_tits)
```

```
['Avatar\\xa0', 'Titanic\\xa0', 'Terminator 2: Judgment Day\\xa0', 'True Lies\\xa0',
 'The Abyss\\xa0', 'Aliens\\xa0', 'The Terminator\\xa0']
```

Deseamos saber qué directores han dirigido el máximo número de películas, junto con ese número de películas.

In [121...

```
# Esta celda debe ser completada por el estudiante

def directors_max_movies_df(
    df_peliculas: pd.DataFrame
) -> tuple[list[str], int]:
    """
    Recibe un dataframe de pandas y devuelve una tupla con el número máximo \
    de películas dirigidas por un director, y una lista con el director (o \
    los directores) que las han dirigido.

    Parameters
    -----
        df_peliculas : El dataframe completo, con todos los datos.

    Returns
    -----
        tuple[list[str], int] : Una tupla con una lista con el/los director/es\
        que han dirigido el número máximo de películas, y ese máximo
    """

    # Excluimos del dataframe original las películas con director desconocido
    directores = df_peliculas[df_peliculas['director_name'] != 'Desc']
    # Contamos cuantas veces aparece cada director en el dataframe
    directores = directores['director_name'].value_counts()
    # Obtenemos el valor máximo de dicho conteo
    max_pelis = directores.max()
    # Obtenemos una lista con el nombre del director (o directores) cuyo
    # conteo sea igual al máximo
    directores_max = directores[directores == max_pelis].index.to_list()

    return (directores_max, max_pelis)
```

In [122...

```
# test de comprobación:

directors_max_movies_df(tabla_breve)
```

Out[122...

```
(['Steven Spielberg'], 26)
```

Parte F. Un cálculo masivo con map-reduce [0,5 puntos]

En este apartado se ha de realizar un programa aparte, *basado en la técnica de map-reduce*, que calcule, para cada idioma, en qué países en que se han producido películas y la suma de los presupuestos de dichas películas. Cuando el idioma o el país o el presupuesto no se conozcan, no se considerará esta película.

```
C:\...> python language_budget_countries.py -q
algunos_campos.txt
```

El programa funcionará necesariamente con la técnica map-reduce, que podemos poner en juego con la librería `mrjob`.

El funcionamiento del mismo se puede activar también desde aquí:

In [123...

```
# Hagamos una llamada al programa de consola desde aquí:  
! python language_budget_countries.py -q algunos_campos.txt
```

```

"Aboriginal"      [["UK", "Australia"], 86000000]
"Arabic"          [["Turkey", "Egypt", "France", "United Arab Emirates"], 11225000]
"Aramaic"         [["USA"], 30000000]
"Bosnian"         [["USA"], 13000000]
"Cantonese"       [["Hong Kong", "China"], 154500000]
"Chinese"         [["China"], 12000000]
"Czech"           [["Czech Republic"], 84450000]
"Danish"          [["Denmark"], 50100000]
"Dari"            [["USA", "Afghanistan"], 20046000]
"Dutch"           [["Netherlands"], 32150000]
"Dzongkha"        [["Australia"], 1800000]
"English"         [["USA", "UK", "New Zealand", "Canada", "Australia", "Germany",
"China", "New Line", "France", "Japan", "Spain", "Hong Kong", "Czech Republic",
"South Korea", "Peru", "Italy", "Aruba", "Denmark", "Libya", "Belgium", "Ireland",
"South Africa", "Switzerland", "Romania", "West Germany", "Chile", "Hungary",
"Russia", "Mexico", "Panama", "Greece", "Netherlands", "Official site", "Bulgaria",
"Iran", "Georgia", "India", "Thailand", "Norway", "Nigeria", "Bahamas", "Iceland",
"Brazil", "Poland", "Kyrgyzstan", "Philippines"], 141889675246]
"Filipino"        [["USA"], 80000000]
"French"          [["France", "Canada", "Finland", "UK"], 1039754500]
"German"          [["Germany", "West Germany", "Switzerland", "Mexico"], 119400000]
"Hebrew"          [["Israel"], 3450000]
"Hindi"           [["India", "USA"], 1027667600]
"Hungarian"       [["Hungary"], 250000000]
"Icelandic"       [["Iceland"], 10000000]
"Indonesian"      [["Indonesia", "UK"], 2100000]
"Italian"         [["UK", "Italy"], 54750000]
"Japanese"        [["Japan", "USA"], 6801919898]
"Kazakh"          [["France"], 25000000]
"Korean"          [["South Korea"], 16449300000]
"Mandarin"        [["China", "Hong Kong", "Taiwan"], 1199632000]
"Maya"            [["USA"], 40000000]
"Mongolian"       [["Russia"], 20000000]
"None"            [["Canada", "USA"], 16500000]
"Norwegian"       [["Norway"], 66500000]
"Panjabi"         [["Canada"], 3000000]
"Persian"         [["Iran"], 690000]
"Portuguese"      [["Brazil"], 26200000]
"Romanian"        [["Romania"], 758000]
"Russian"         [["Soviet Union", "Russia"], 133700000]
"Slovenian"       [["Slovenia"], 500000]
"Spanish"         [["USA", "Mexico", "Spain", "Italy", "Argentina", "Colombia", "Dominican Republic"], 951325775]
"Swahili"         [["Kenya"], 15000]
"Swedish"         [["Sweden"], 50400000]
"Tamil"           [["India"], 150000000]
"Telugu"          [["India"], 18026148]
"Thai"            [["Thailand"], 900000000]
"Urdu"            [["Pakistan"], 1000000]
"Vietnamese"      [["USA"], 1592000]
"Zulu"            [["UK", "South Africa"], 13000000]

```

In [124...

Para que el resultado se almacene en un archivo:

! python language_budget_countries.py -q algunos_campos.txt > language_countries

La siguiente celda me permite ver tu programa cómodamente desde aquí.

In [125...

```
def print_file(filename):
    with open(filename, "r") as f:
        for line in f:
            print(line, end="")

print_file("language_budget_countries.py")

import sys
from mrjob.job import MRJob
from collections import defaultdict

class MRLanguageCountriesBudgets(MRJob):
    def mapper(self, _, linea):
        # Para cada línea, obtenemos sus componentes, que están separados por |
        titulo, anho, idioma, pais, presupuesto = linea.split('|')

        # Intentamos convertir a int el año y el presupuesto
        try:
            anho = int(anho)
            presupuesto = int(presupuesto)
        except:
            pass

        # Si se conocen los datos de la película (es decir, existen el idioma y
        # país y el presupuesto y año existen o son distintos de -1)
        if idioma and pais and presupuesto and anho \
            and presupuesto != -1 and anho != -1:
            # Generamos los pares de clave-valor, con idioma como clave, y
            # el país y presupuesto como valor
            yield idioma, (pais, int(presupuesto))

    def reducer(self, idioma, values):
        # Generamos un defaultdict para el presupuesto
        pais_presupuesto = defaultdict(int)
        # Iteramos sobre los valores recibidos del mapper
        for pais, presupuesto in values:
            # Sumamos el presupuesto de cada país
            pais_presupuesto[pais] += presupuesto
        # Producimos los pares clave-valor, donde la clave es el idioma, y
        # los valores una lista con cada país y la suma de los presupuestos
        yield idioma, (list(pais_presupuesto.keys()),
                       sum(pais_presupuesto.values()))

# Para permitir la ejecución por línea de comandos
if __name__ == '__main__':
    MRLanguageCountriesBudgets.run()
```

Parte G. Un apartado libre [0.5 puntos]

En este último apartado, se propone obtener diferentes estadísticas a partir de un archivo CSV obtenido de internet con diferentes estadísticas y datos de los jugadores de la NBA desde la temporada 1996-97 hasta la temporada 2022-23.

G1. Estadísticas

En primer lugar, importaremos el archivo con pandas y obtendremos distintas estadísticas sencillas. La primera tarea será obtener el promedio del peso y altura por equipo en la temporada 22-23, con dos cifras decimales, y ordenar los datos de forma descendiente por altura y peso.

In [126...

```
# Guardamos el archivo como un csv,  
# indicando que use la primera columna como índice  
jugadores = pd.read_csv(OPTIONAL_DATA, sep=',', index_col=0)  
  
# Obtenemos los registros sólo de la temporada actual  
jugadores_actual = jugadores[jugadores['season'] == '2022-23']  
  
# Para obtener la media, agrupamos en primer lugar por equipo, y luego  
# calculamos la media de la altura y del peso, redondeada a dos decimales  
media_altura_peso = jugadores_actual.groupby('team_abbreviation')[  
    ['player_height', 'player_weight']].mean().round(2)  
  
# Ordenamos los valores por orden descendiente de altura y, en caso de  
# empate, por peso  
media_altura_peso.sort_values(by=['player_height', 'player_weight'],  
                              ascending=False)
```


Out[126...

	player_height	player_weight
team_abbreviation		
ORL	202.15	101.26
UTA	201.30	103.31
BOS	200.80	102.44
MIL	200.52	99.94
HOU	200.49	100.76
DEN	200.34	102.34
TOR	200.10	96.44
OKC	200.02	94.35
SAC	199.90	98.18
NOP	199.71	96.36
CHA	199.61	93.04
SAS	199.57	98.41
WAS	199.52	93.69
MIA	199.46	100.75
DET	199.46	98.45
POR	199.33	95.88
MIN	199.11	99.03
PHI	198.97	100.04
ATL	198.97	96.04
BKN	198.88	98.82
CHI	198.87	97.04
CLE	198.57	99.15
PHX	198.44	98.46
LAL	198.40	94.52
LAC	198.12	95.10
IND	198.12	94.85
NYK	197.96	98.57
DAL	197.64	95.54
GSW	196.92	94.83
MEM	196.85	98.10

A continuación, analizaremos el rendimiento de los jugadores según su posición en el draft. Para ello, se considerará el rendimiento como el promedio de los puntos anotados, rebotes y asistencias.

In [127...

```
jugadores_draft = jugadores
# Convertimos el 'draft_number' a string, para facilitar el trabajo
jugadores_draft['draft_number'] = jugadores_draft['draft_number'].astype(str)
# Con .loc, accedemos a las filas que cumplan la condición interna. En este
# caso, indicamos que queremos obtener, para las filas donde 'draft_number' es
# 0, ese draft_number, el cual posteriormente cambiaremos por 'Undrafted'. Esto
# es porque en el archivo se usa 0 y Undrafted para los jugadores no elegidos,
# y debemos unificarlo todo bajo la misma nomenclatura
jugadores_draft.loc[
    jugadores_draft['draft_number'] == '0', 'draft_number'
] = 'Undrafted'

# Agrupamos por draft_number y hallamos la media de puntos anotados, rebotes
# cogidos y asistencias dadas
rend_por_pos = jugadores_draft.groupby('draft_number')[
    ['pts', 'reb', 'ast']
].mean()
# Añadimos una columna rendimiento, como el promedio de las tres anteriores
rend_por_pos['rendimiento'] = rend_por_pos.mean(axis=1).round(2)

# Imprimimos los resultados, ordenamos por rendimiento de forma descendente
rend_por_pos[['rendimiento']].sort_values('rendimiento', ascending=False)
```

Out[127...

rendimiento	
draft_number	
1	8.87
3	7.86
2	7.03
5	6.98
4	6.94
...	...
82	1.40
79	1.03
120	1.00
165	0.58
124	0.52

75 rows × 1 columns

Por último, se pide hallar el número total de jugadores que han jugado en la NBA provenientes de cada país, y devolver el conteo como un diccionario.

In [128...

```
# Reducimos el dataframe a los jugadores y el país, y eliminamos duplicados
jugadores_unicos = jugadores[['player_name', 'country']].drop_duplicates()
# Obtenemos una tabla con cada país, y un recuento del total de jugadores
# de ese país. Indicamos que resetee los índices de la tabla
jugadores_pais = jugadores_unicos['country'].value_counts().reset_index()
# Cambiamos el nombre de las columnas
jugadores_pais.columns = ['country', 'total_players']

# Establecemos el índice del diccionario como el país, seleccionamos la
# columna total players y lo convertimos a diccionario
dict_jug_pais = jugadores_pais.set_index('country')['total_players'].to_dict()

print(dict_jug_pais)
```

```
{'USA': 2140, 'Canada': 45, 'France': 37, 'Australia': 22, 'Croatia': 15, 'Serbia': 15, 'Argentina': 14, 'Spain': 14, 'Germany': 13, 'Brazil': 13, 'Turkey': 12, 'Lithuania': 12, 'Nigeria': 12, 'Senegal': 10, 'Slovenia': 10, 'Russia': 9, 'Greece': 9, 'Ukraine': 8, 'Italy': 7, 'Dominican Republic': 6, 'Georgia': 6, 'Puerto Rico': 6, 'Latvia': 6, 'Montenegro': 5, 'Serbia and Montenegro': 5, 'Jamaica': 5, 'Cameroon': 4, 'Czech Republic': 4, 'Yugoslavia': 4, 'China': 4, 'Democratic Republic of the Congo': 4, 'United Kingdom': 4, 'Poland': 4, 'DRC': 3, 'Japan': 3, 'Bosnia and Herzegovina': 3, 'Bahamas': 3, 'South Sudan': 3, 'Israel': 3, 'Mexico': 3, 'Netherlands': 2, 'Congo': 2, 'Bosnia': 2, 'Sweden': 2, 'Mali': 2, 'Haiti': 2, 'Venezuela': 2, 'Switzerland': 2, 'Gabon': 2, 'New Zealand': 2, 'Finland': 2, 'England': 2, 'Saint Lucia': 1, 'Sudan': 1, 'Republic of the Congo': 1, 'Angola': 1, 'Guinea': 1, 'Trinidad and Tobago': 1, 'Denmark': 1, 'Egypt': 1, 'U.S. Virgin Islands': 1, 'Colombia': 1, 'Austria': 1, 'Ghana': 1, 'Macedonia': 1, 'Tunisia': 1, 'Sudan (UK)': 1, 'Scotland': 1, 'Ireland': 1, 'St. Vincent & Grenadines': 1, 'South Korea': 1, 'USSR': 1, 'Uruguay': 1, 'Iran': 1, 'Cabo Verde': 1, 'Belize': 1, 'Tanzania': 1, 'Panama': 1, 'Great Britain': 1, 'US Virgin Islands': 1, 'Bosnia & Herzegovina': 1, 'Portugal': 1}
```

G2. Gráficos

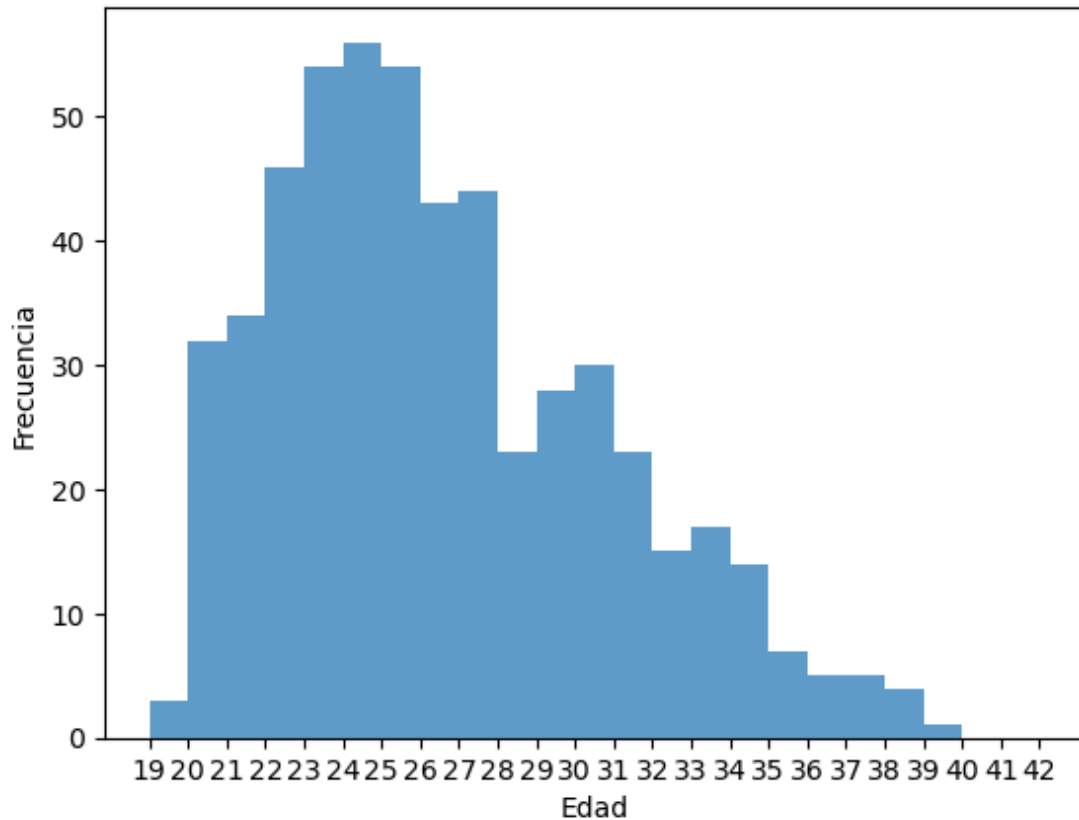
En primer lugar, se propone crear un gráfico que represente un histograma de la edad de los jugadores de la temporada actual.

In [129...

```
# Generamos un rango desde la edad mínima hasta la máxima
rango = range(int(jugadores_actual['age'].min()),
              int(jugadores_actual['age'].max()))

# Representamos el histograma con el rango como bins
plt.hist(jugadores_actual['age'], bins=rango, alpha=0.7)
plt.title('Distribución de las edades de los jugadores en la temporada actual')
plt.xlabel('Edad')
plt.ylabel('Frecuencia')
plt.xticks(rango)
plt.show()
```

Distribución de las edades de los jugadores en la temporada actual



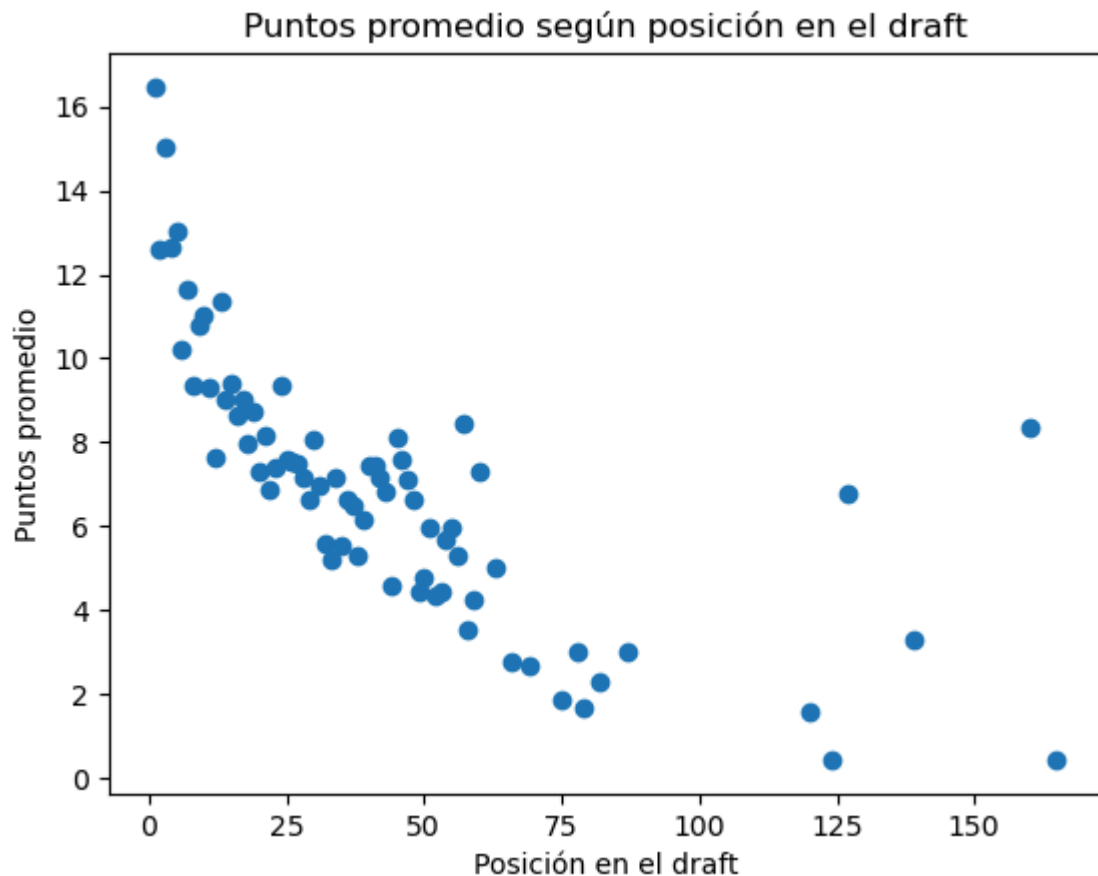
El siguiente será un gráfico de puntos que relacione la posición en el draft con los puntos promedios anotados por el jugador elegido en esa posición.

In [130...

```
# Calculamos el promedio de puntos anotados por posición en el draft
puntos_draft = (
    jugadores_draft
    .groupby('draft_number')['pts']
    .mean()
    .round(2)
    .reset_index()
)
# Excluimos a los jugadores que no obtuvieron posición en el draft
puntos_draft = puntos_draft[puntos_draft['draft_number'] != 'Undrafted']
# Convertimos de nuevo 'draft_number' a tipo int
puntos_draft['draft_number'] = pd.to_numeric(puntos_draft['draft_number'])

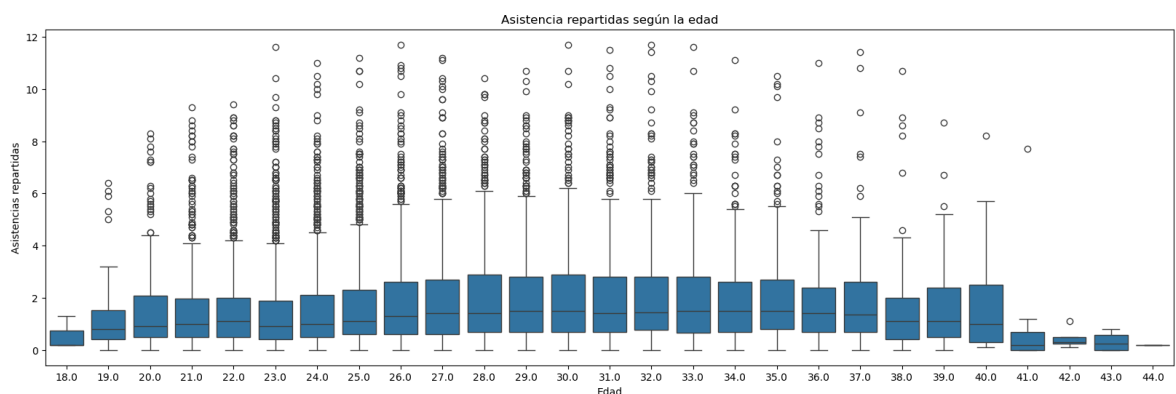
# Generamos el scatter plot, con el eje X como el 'draft_number'
# y el eje Y como los puntos promedio
plt.scatter(puntos_draft['draft_number'], puntos_draft['pts'])
plt.title('Puntos promedio según posición en el draft')
plt.xlabel('Posición en el draft')
plt.ylabel('Puntos promedio')

plt.show()
```



A continuación, crearemos un diagrama de cajas, que nos permita analizar cómo varían las asistencias que da un jugador con la edad, para analizar si hay algún tipo de relación. Esto nos permitirá también explorar nuevas librerías para visualización de datos que no habíamos usado anteriormente.

```
In [131... plt.figure(figsize=(20,6))
# Establecemos la edad en el eje X y las asistencias repartidas en el Y,
# indicando que tome los datos del dataset jugadores
sns.boxplot(x='age', y='ast', data=jugadores)
plt.xlabel('Edad')
plt.ylabel('Asistencias repartidas')
plt.title('Asistencia repartidas según la edad')
plt.show()
```

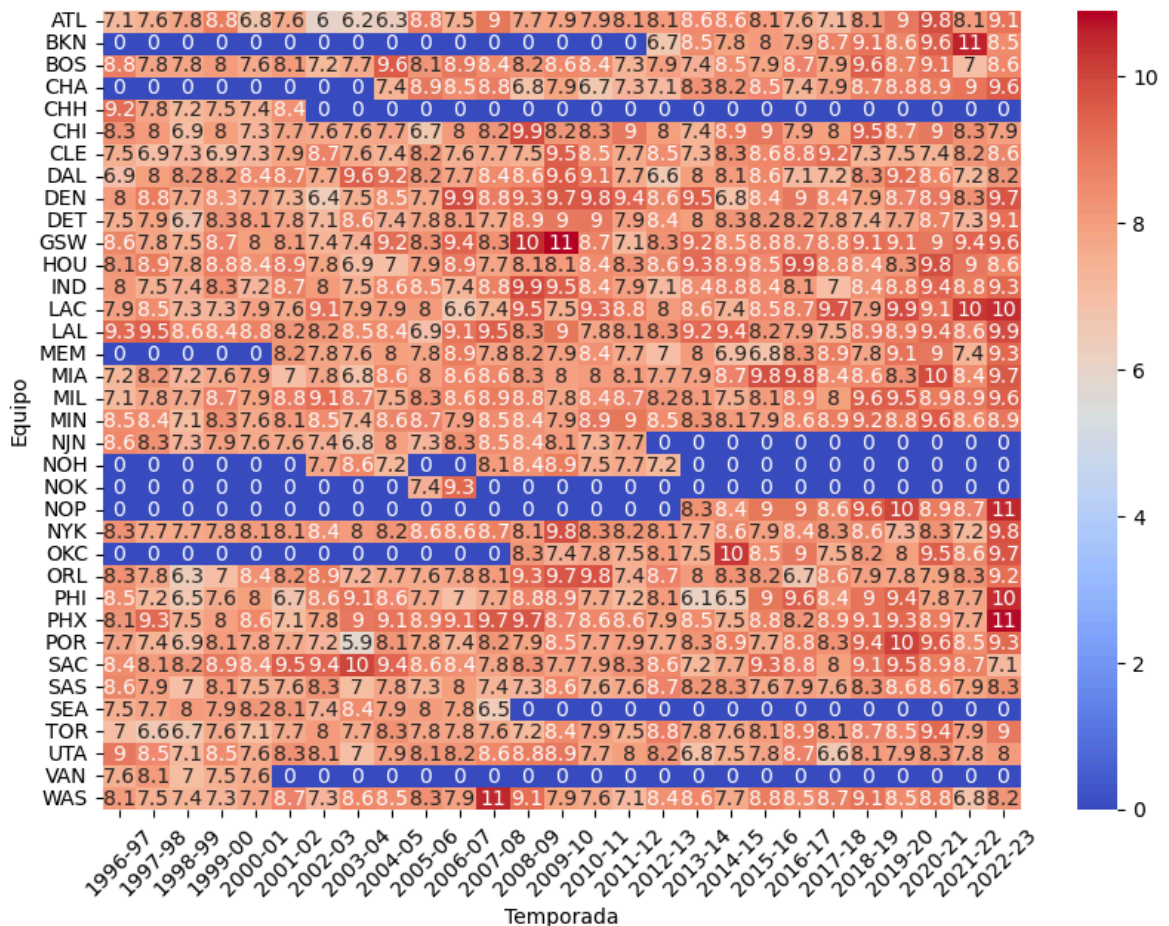


Por último, para terminar de testear las funcionalidades de la librería seaborn, crearemos un heatmap para visualizar la media de puntos anotados por cada equipo en cada temporada.

In [132...

```
# Creamos una tabla con el equipo como índice, una columna por temporada
# y los valores de cada celda serán el promedio de la columna 'pts',
# rellenando con 0 los valores faltantes
media_puntos = jugadores.pivot_table(
    index='team_abbreviation',
    columns='season',
    values='pts',
    aggfunc='mean',
    fill_value=0
)

plt.figure(figsize=(10,7))
# Creamos el heatmap, mostramos los valores con annot=True, y establecemos
# el esquema de color con cmap como una escala de rojo a azul
sns.heatmap(media_puntos, annot=True, cmap='coolwarm')
plt.xlabel('Temporada')
plt.ylabel('Equipo')
plt.xticks(rotation=45)
plt.show()
```



G3. Map-reduce

Como ejercicio de map-reduce, propongo calcular el MVP de cada temporada. Para ello, se asignará un peso distinto a cada una de las estadísticas, siendo del 50% los puntos anotados, un 30% los rebotes cogidos y un 20% las asistencias dadas, añadiendo el net_rating como un bonus del 10% (de forma que, si es positivo, suma; y si es negativo, resta). Este map-reduce se ejecutará desde el archivo `mvp_seasons.py`. Previamente, se habrá creado un archivo de texto con los campos de interés, que en este caso son el

nombre del jugador, la temporada, su equipo y los datos estadísticos (puntos, rebotes, asistencias, rating).

```
In [133... def generaArchivoInteres(datos_jugadores: pd.DataFrame, arch_out: str) -> None:
    """
    Recibe un dataframe de pandas y la url de un archivo de destino, y \
    extrae algunos campos de interés del dataframe y los imprime al archivo, \
    separados por comas.

    Parameters
    -----
        datos_jugadores : El dataframe completo, con todos los datos.
        arch_out : La ruta al archivo que se creará con los datos reducidos.

    Returns
    -----
        None
    """

    campos = ['player_name', 'season', 'team_abbreviation',
              'pts', 'reb', 'ast', 'net_rating']
    df_interes = datos_jugadores[campos] # Obtenemos los campos de interés

    with open(arch_out, 'w') as archivoSalida:
        # Iteramos sobre las filas del dataframe
        for indice, linea in df_interes.iterrows():
            # Concatenamos cada valor con una coma, y añadimos
            # un salto de línea al final de todos ellos
            archivoSalida.write(','.join(map(str, linea.values))+'\n')

    archivoSalida.close()

generaArchivoInteres(jugadores, 'datos_interes.txt')
```

```
In [134... ! python mvp_seasons.py -q datos_interes.txt
```

"1996-97"	[40.786, "Michael Jordan"]
"1997-98"	[46.208, "Shaquille O'Neal"]
"1998-99"	[30.925, "Karl Malone"]
"1999-00"	[41.349, "Shaquille O'Neal"]
"2000-01"	[34.253, "Tim Duncan"]
"2001-02"	[37.432, "Shaquille O'Neal"]
"2002-03"	[36.915, "Dirk Nowitzki"]
"2003-04"	[35.231, "Kevin Garnett"]
"2004-05"	[37.433, "Tim Duncan"]
"2005-06"	[31.302, "Dwyane Wade"]
"2006-07"	[33.022, "Dirk Nowitzki"]
"2007-08"	[34.668, "Kevin Garnett"]
"2008-09"	[44.262, "LeBron James"]
"2009-10"	[39.021, "LeBron James"]
"2010-11"	[34.85, "LeBron James"]
"2011-12"	[35.521, "LeBron James"]
"2012-13"	[41.597, "LeBron James"]
"2013-14"	[34.776, "Kevin Durant"]
"2014-15"	[39.771, "Stephen Curry"]
"2015-16"	[50.968, "Stephen Curry"]
"2016-17"	[41.67, "Stephen Curry"]
"2017-18"	[39.0, "Naz Mitrou-Long"]
"2018-19"	[62.8, "RJ Hunter"]
"2019-20"	[50.673, "Giannis Antetokounmpo"]
"2020-21"	[39.578, "Joel Embiid"]
"2021-22"	[36.973, "Jayson Tatum"]
"2022-23"	[39.938, "Nikola Jokic"]

Datos personales

- **Apellidos:** Gómez Sabucedo
- **Nombre:** Hugo
- **Email:** hugogomezsabucedo@gmail.com
- **Fecha:** 14 de noviembre de 2024

Ficha de autoevaluación

Creo que, en general, he conseguido resolver todos los ejercicios que se planteaban en la tarea de forma satisfactoria, obteniendo los resultados esperados. En algunos apartados, la solución a los mismos puede que no sea la más eficiente, a nivel de código, de velocidad, o no estar completamente en estilo pitónico. Es por eso que considero que, si bien el resultado general de la práctica es satisfactorio, no alcanza la nota completa.

Apartado	Calificación	Comentario
a)	1.75 / 2.0	Completamente resuelto
b)	1.75 / 2.0	Completamente resuelto
c)	1.0 / 1.0	Completamente resuelto

Apartado	Calificación	Comentario
d)	1.75 / 2.0	Completamente resuelto
e)	1.75 / 2.0	Completamente resuelto
f)	0.5 / 0.5	Completamente resuelto
g)	0.5 / 0.5	Completamente resuelto
Total	9.0 / 10.0	Notable

Ayuda recibida y fuentes utilizadas

- Microsoft Copilot: como herramienta de apoyo y consulta
- <https://peps.python.org/pep-0008/> : para consultar el estilo pep8
- <https://medium.com/databale/beginners-guide-for-mapreduce-with-mrjob-in-python-dbd2e7dd0f86> : para aprender sobre el uso del mapreduce
- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> : para documentación sobre BeautifulSoup
- <https://www.w3schools.com/python/> : con documentación y ejemplos generales y de ejercicios de Python
- <https://www.geeksforgeeks.org/python-programming-language-tutorial/> : con ejemplos de usos de Python
- <https://stackoverflow.com/> : para alguna consulta o duda más concreta
- <https://www.kaggle.com/datasets/justinas/nba-players-data> : la fuente del dataset utilizado

```
In [ ]: # Esta celda se ha de respetar: está aquí para comprobar
        # el funcionamiento de algunas funciones por parte de tu profesor
```