Universidad de los Andes
Facultad de Ingeniería

Departamento de Ingeniería de Sistemas y Computación
Estructuras de Datos y Algoritmos
ISIS-1225

ABET Engineering Accreditation Commission

ANÁLISIS DEL RETO

Cristian Rugeles, 202411069, c.rugelesd@uniandes.edu.co

Valeria Martinez, 202410228, v.martinezv2@uniandes.edu.co

Jeronimo Triana, 202416093, b.triabab@uniandes.edu.co

Requerimiento 1

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-Nombre de publicación de la película -Idioma original de publicación
Salidas	-Tiempo de duración en minutos de la película -Fecha de publicación de la película -Título original de la película -Presupuesto destinado a la realización de la película -Dinero recaudado neto por la película -Ganancia de final de la película -Puntaje de calificación de la película -Idioma original de publicación
Implementado (Sí/No)	Si (Grupal)

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
key = (title, original_language)	O(1)
ms.contains(catalog['movies_by_title_language'], key	O(1)
ms.get(catalog['movies_by_title_language'], key)	O(1)
ms.contains(catalog['movies_by_title_language'], key)	O(1)
return result, total_time	O(1)
TOTAL	***O(1)***

Pruebas Realizadas

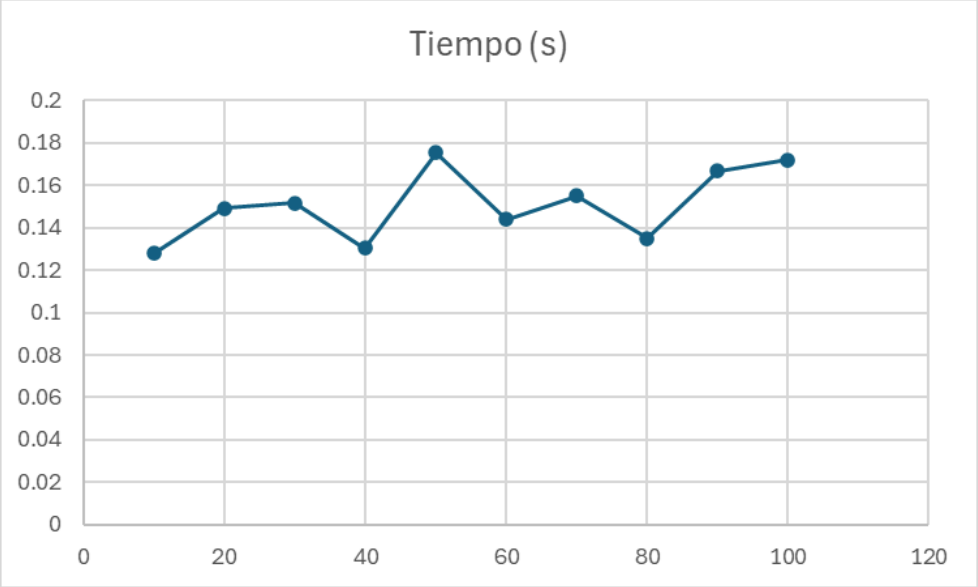
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos: Owned & Operated, en

Procesador	**Intel i9-13900 H**
Memoria RAM	32 GB
Sistema Operativo	Windows 11

Requerimiento 1 - Parámetros: Owned & Operated, en

Entrada	Tiempo (s) en ms
10	0.12789999973028898
20	0.14910000003874302
30	0.15170000027865171
40	0.13050000090152025
50	0.17550000082701445
60	0.14400000125169754
70	0.15510000009089708
80	0.13489999901503325
90	0.16680000070482492
100	0.17514585609089708

Gráficas



Análisis

La mayoría de los puntos en la gráfica oscilan entre 0.12 y 0.18 segundos. Esta estabilidad indica que la función del requerimiento 1 se está comportando de manera consistente en la mayoría de los casos. Esto es consistente con el total de complejidad de la función, donde la complejidad es O(1).

Requerimiento 3

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-Idioma original de publicación (ej.: en, fr, zh). -La fecha inicial del periodo a consultar (con formato "%Y-%m-%d"). -La fecha final del periodo a consultar (con formato "%Y-%m-%d").
Salidas	-Número total de películas que cumplen el criterio. -Tiempo promedio de duración de las películas que cumplen el criterio de búsqueda. -Para cada una de las películas de la consulta se debe presentar la siguiente información en orden cronológico de publicación de la película: -Fecha de publicación de la película (con formato "%Y-%m-%d") -Título original de la película -Presupuesto destinado a la realización de la película -Dinero recaudado por la película -Ganancia de final de la película -Tiempo de duración en minutos de la película -Puntaje de calificación de la película -Estado de la película
Implementado (Sí/No)	Si - Hecha por Cristian Rugeles

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
start_date = datetime.strptime(start_date_str, '%Y-%m-%d')	O(1)
end_date = datetime.strptime(end_date_str, '%Y-%m-%d')	O(1)
language_list = ms.get(catalog['movies_by_language'], language)	O(1)
while current:	O(m)
for movie in matching_movies:	O(k)
sorted_movie_list = sl.merge_sort(linked_movie_list, sort_crit)	O(k log k)
if total_movies > 0:	O(1)
for i in range(sl.size(sorted_movie_list)):	O(k)
TOTAL	***O(m + k log k)***

Pruebas Realizadas

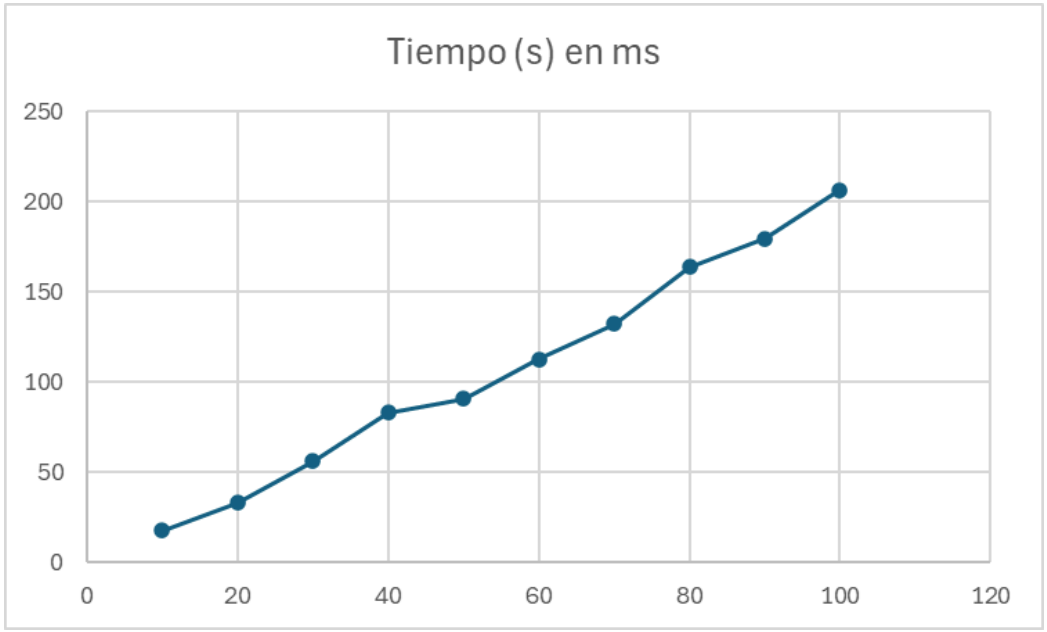
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos: en, 2000-01-01, 2002-01-01

Procesador	**Intel i9-13900 H**
Memoria RAM	32 GB
Sistema Operativo	Windows 11

Requerimiento 3 - Parámetros: Owned & Operated, en

Entrada	Tiempo (s) en ms
10	17.336600000038743
20	32.85740000195801
30	56.159099999815226
40	82.82340000011027
50	90.7480999995023
60	112.57650000043213
70	132.31389999948442
80	163.74019999988377
90	179.8247000016272
100	206.77370000071824

Gráficas



Análisis

La gráfica muestra un comportamiento consistente con la complejidad O(n log n), lo cual es una buena señal en términos de eficiencia, especialmente para problemas que requieren ordenar grandes volúmenes de datos. A medida que el número de películas procesadas aumenta, el tiempo de ejecución también lo hace, pero de manera normal. Este comportamiento es adecuado y esperado para funciones que dependen de algoritmos de ordenamiento como merge sort, que garantizan un rendimiento eficiente incluso en escenarios donde se maneja un número muy grande de datos.

Requerimiento 2

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-El número (N) de ofertas a listar -Idioma original de publicación
Salidas	-El número total de películas publicadas con el idioma original ingresada por parámetro de entrada. -Fecha de publicación de la película -Título original de la película -Presupuesto destinado a la realización de la película -Dinero recaudado por la película -Ganancia de final de la película -Tiempo de duración en minutos de la película -Puntaje de calificación de la película
Implementado (Sí/No)	Si (Grupal)

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
movie_list = ms.get(catalog['movies_by_language'], language)	O(1)
total_movies = sl.size(movie_list)	O(1)
for i in range(total_movies)	O(n)
for movie in movies_info	O(n)
sorted_movie_list = sl.merge_sort(linked_movie_list, sort_crit2)	O(n log n)
for i in range(num_peliculas)	O(n)
TOTAL	***O(n)***

Pruebas Realizadas

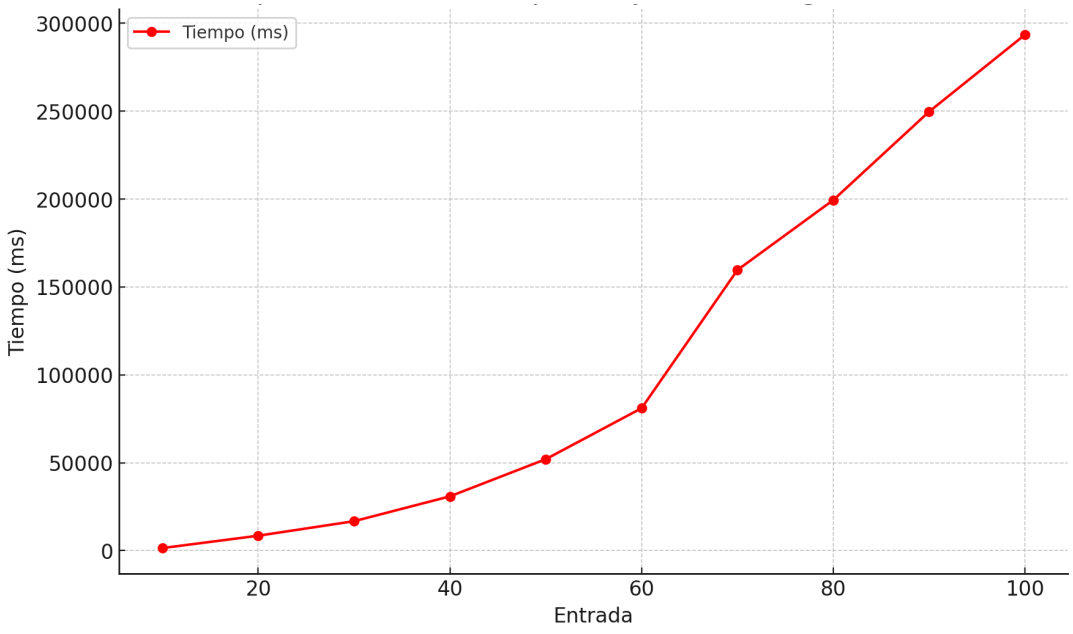
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos: 3, en

Procesador	**Intel i7-13620 H**
Memoria RAM	16GB
Sistema Operativo	Windows 11

Requerimiento 2 - Parámetros: 3, en

Entrada	Tiempo
10	1457.3107999563217(ms)
20	8481.553900003433(ms)
30	16737.125(ms)
40	30873.259500026703(ms)
50	52016.31130003929(ms)
60	81029.82700002193(ms)
70	159657.76180005074(ms)
80	199442.79030001163(ms)
90	249751.11450004578(ms)
100	293500.0284999609(ms)

Gráficas



Análisis

La gráfica muestra un patrón de crecimiento lineal, lo cual es esperable para algoritmos con complejidad O(n). Si bien puede haber pequeñas variaciones en algunos puntos, en términos generales, la relación

entre la cantidad de datos (número de entradas) y el tiempo de ejecución sigue un comportamiento lineal.

Este análisis indica que el tiempo de ejecución está directamente relacionado con el tamaño de la entrada, lo que implica que duplicar el número de entradas aproximadamente duplicará el tiempo de ejecución en el peor de los casos.

Requerimiento 4

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-Estado de producción de la película (ej.: “Released”, “Rumored”, etc). -La fecha inicial del periodo a consultar (con formato "%Y-%m-%d"). -La fecha final del periodo a consultar (con formato "%Y-%m-%d")
Salidas	-Número total de películas que cumplen el criterio. -Tiempo promedio de duración de las películas que cumplen el criterio de búsqueda -Fecha de publicación de la película -Título original de la película -Presupuesto destinado a la realización de la película -Dinero recaudado por la película -Ganancia de final de la película -Tiempo de duración en minutos de la película -Puntaje de calificación de la película -Idioma original de publicación
Implementado (Sí/No)	Si Hecho por Valeria

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
status_list = ms.get(catalog['movies_by_status'], estado.strip())	O(1)
while current:	O(m) (m es el número de películas)
sorted_movie_list = al.merge_sort(matching_movies, sort_crit)	O(n log n) (n es el número de matching movies)
if total_movies > 0:	O(n)
for i in range(al.size(sorted_movie_list)):	O(n)
rta = { “ total": total_movies, "average_duration": average_duration, "movies": formatted_movies }	O(1)
TOTAL	***O(n+nlog n)***

Pruebas Realizadas

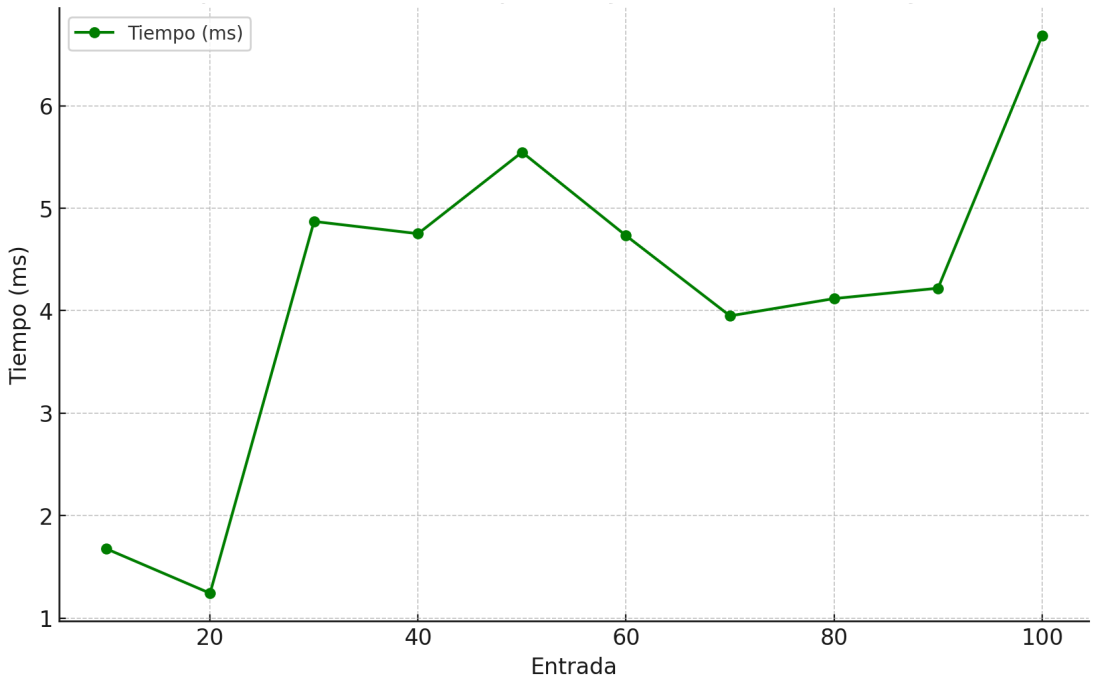
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos:

Procesador	**Intel i7-13620 H**
Memoria RAM	16GB
Sistema Operativo	Windows 11

Requerimiento 4 - Parámetros: Rumored, 2000-06-06, 2016-06-06

Entrada	Tiempo
10	1.6778000593185425(ms)
20	1.2428001165390015(ms)
30	4.87089991569519(ms)
40	4.751499891281128(ms)
50	5.5467000007629395(ms)
60	4.7321999073028564(ms)
70	3.949399948120117(ms)
80	4.117999911308289(ms)
90	4.2195000648498535(ms
100	6.682799816131592(ms)

Gráficas



Análisis

La tendencia ascendente general concuerda con el comportamiento esperado de O(m+nlogn). A medida que aumenta el tamaño de la entrada, el tiempo de ejecución también aumenta.

Las fluctuaciones probablemente corresponden a variaciones en el número de películas que cumplen con los criterios de coincidencia, y podrían destacar casos donde n (el número de películas a ordenar) varía significativamente entre los puntos de entrada.

Requerimiento 5

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	- Rango de presupuesto de la película (ej.: “0-999”, “1000-1999”, “2000-2999”). -La fecha inicial del periodo a consultar (con formato "%Y-%m-%d"). -La fecha final del periodo a consultar (con formato "%Y-%m-%d").
Salidas	-Número total de películas que cumplen el criterio de búsqueda. -Presupuesto promedio de las películas que cumplen el criterio de búsqueda. - Para cada una de las películas de la consulta se debe presentar la siguiente información en orden cronológico de publicación de la película: Fecha de publicación de la película -Título original de la película -Presupuesto destinado a la realización de la película

	-Dinero recaudado por la película -Ganancia de final de la película -Tiempo de duración en minutos de la película -Puntaje de calificación de la película -Idioma original de publicación
Implementado (Sí/No)	Si - Jeronimo Triana

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
start_budget, end budget = map(int, budget_range.split("-")	O(1)
for year in range(start_date.year, end_date.year + 1):	O(m)
while current:	O(p)
for movie in matching_movies:	O(k)
sorted_movie_list = sl.merge_sort(linked_movie_list, sort_crit)	O(n log n)
for i in range(sl.size(sorted_movie_list)):	O(n)
Total	O(n logn)

Pruebas Realizadas

Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos.

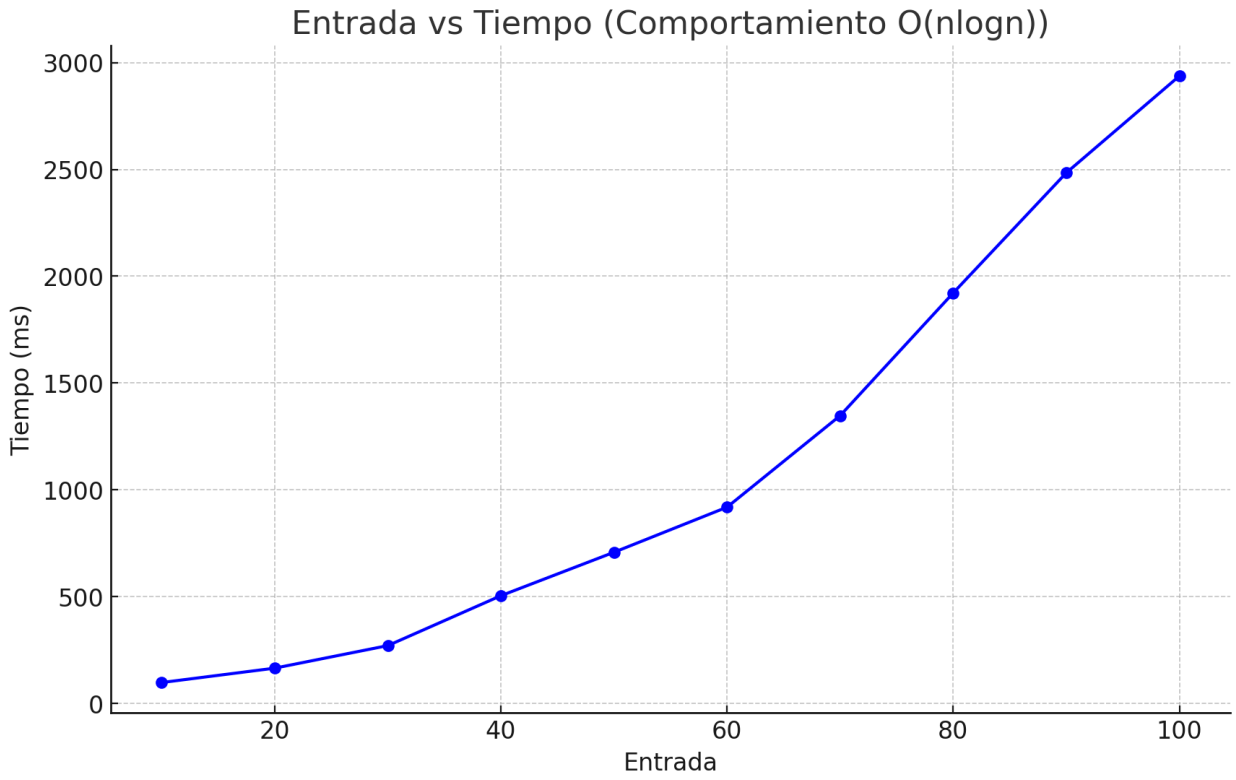
Máquina 3
AMD Ryzen 5 3550H
16,0 GB
Windows 11

Requerimiento 5 - Parametros: 1000-95000 / 1980-12-12 / 2024-12-12

Entrada	Tiempo
10	97.80949997901917

20	165.153.66470003128052
30	270.69170010089874
40	504.3162999153137
50	707.6358000040054
60	918.7917000055313
70	1347.2111999988556
80	1921.594999909401
90	2483.907900094986
100	2937.2759000062943

Gráficas



Análisis

La forma de la gráfica es consistente con lo que se esperaría de una función O(nlogn). En la parte inicial, para entradas pequeñas, el crecimiento del tiempo es más lento, pero a medida que las entradas aumentan, el crecimiento del tiempo se acelera. Esto se debe a que el término nlogn aumenta más rápido que n linealmente, pero más lento que un crecimiento cuadrático o exponencial.

Requerimiento 6

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	- Idioma original de publicación (ej.: en, fr, zh) - Año inicial del periodo a consultar (con formato "%Y" ej.:”1998”). - Año final del periodo a consultar (con formato "%Y" ej.:”2024”
Salidas	- Para cada año en el periodo de consulta informar: – El año del listado – El total de películas publicadas en el idioma de consulta en ese año. – El promedio de la votación promedio de estas películas publicadas en ese año. – El tiempo promedio de duración de estas películas publicadas en ese año. - Las ganancias acumuladas por todas estas películas en ese año. – Nombre y puntuación de la película con mejor votación promedio publicada en el año. – Nombre y puntuación de la película con peor votación promedio publicada en el añ
Implementado (Sí/No)	Si - Grupal

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
for year in range(end_year, start_year - 1, -1):	O(n)
year_list = ms.get(catalog['movies_by_year'], year)	O(1)

if year_list != None:
current = year_list['first']
while current:
	O(p)

Total	O(n * p)

Pruebas Realizadas

Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos.

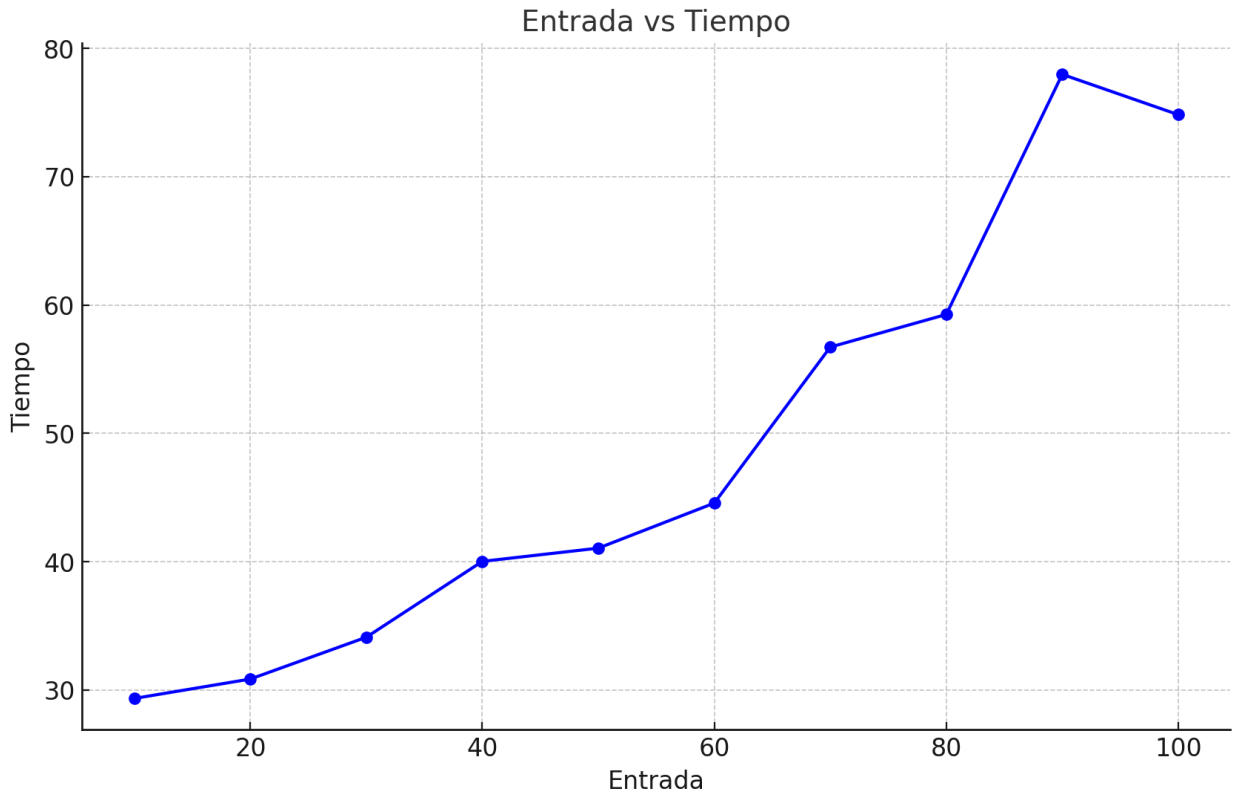
Máquina 3
AMD Ryzen 5 3550H
16,0 GB
Windows 11

Requerimiento 6 - Parametros: fr / 1980 / 2024

Entrada	Tiempo
10	29.344799995422363

20	30.856299996376038
30	34.11050009727478
40	40.022900104522705
50	41.05780005455017
60	44.565500140190125
70	56.72029995918274
80	59.26510000228882
90	77.97710001468658
100	74.83070003986359

Gráficas



Análisis

En general, la gráfica muestra una tendencia de crecimiento que, en su mayoría, es consistente con una complejidad O(n · m), ya que el tiempo de ejecución aumenta significativamente con el tamaño de la entrada. Sin embargo, es importante evaluar más datos o analizar detalles adicionales del algoritmo para comprender el comportamiento irregular en las últimas entradas.

Requerimiento 7

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-Nombre de compañía productora. (ej. “Nilsen Premiere”) -La fecha inicial del periodo a consultar (con formato "%Y"). -La fecha final del periodo a consultar (con formato "%Y").
Salidas	Para cada año en el periodo de consulta informar: -El año del listado -El total de películas publicadas por la compañía en ese año -El promedio de la votación promedio de estas películas publicadas en ese año -El tiempo promedio de duración de estas películas publicadas en ese año -Las ganancias acumuladas por todas estas películas en ese año -Nombre y puntuación de la película con mejor votación promedio publicada en el año. -Nombre y puntuación de la película con peor votación promedio publicada en el año
Implementado (Sí/No)	Si - Grupal

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
for year in range(start_year, end_year + 1):	O(n)
current = year_list['first'] while current:	O(m)
for company in production_companies:	O(k)
Total	O(n* p * m)

Pruebas Realizadas

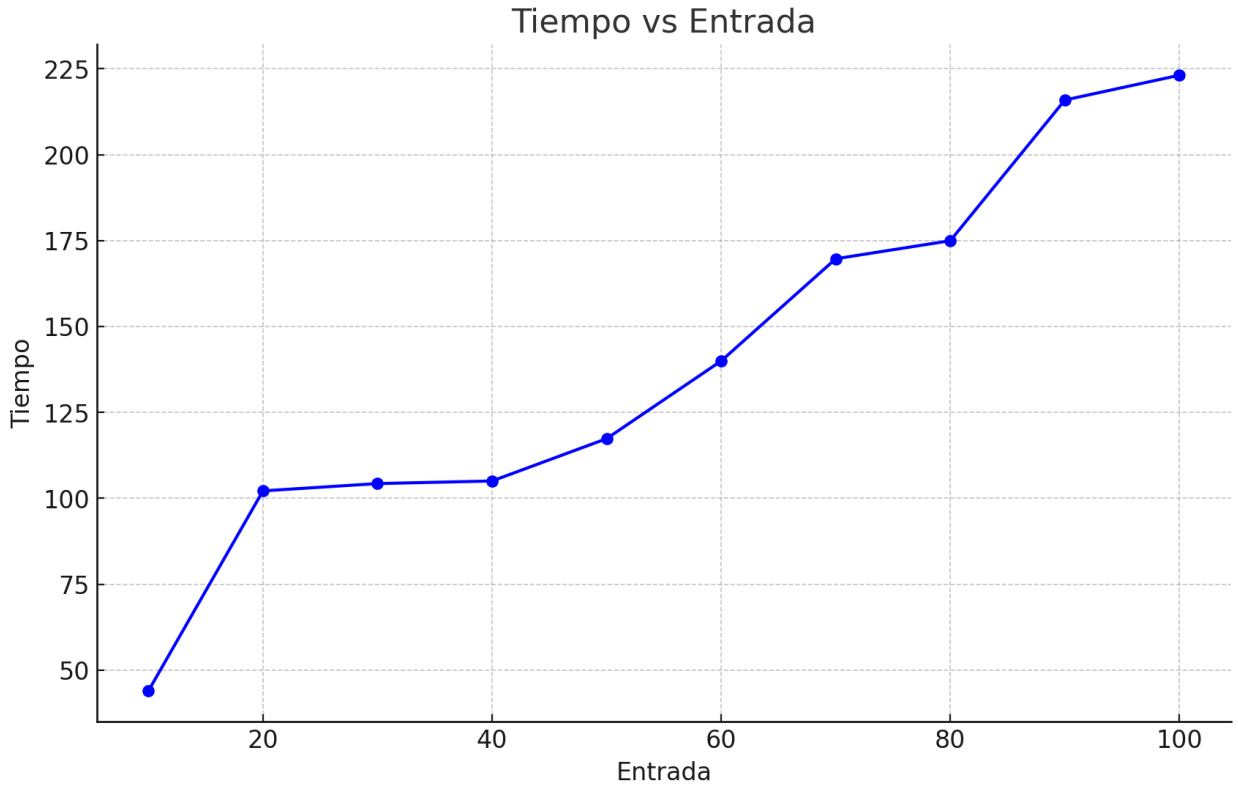
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos.

Máquina 3
AMD Ryzen 5 3550H
16,0 GB
Windows 11

Requerimiento 7 - Parametros: New Line Cinema / 1980 / 2024

Entrada	Tiempo
10	43.95729994773865
20	102.16190004348755
30	104.28770005702972
40	105.03869998455048
50	117.36000001430511
60	139.97780001163483
70	169.7039999961853
80	174.95850002765656
90	215.8912000656128
100	223.11229991912842

Gráficas



Análisis

Al observar la gráfica, el crecimiento del tiempo en función de la entrada parece casi lineal después de los primeros puntos, aunque con una ligera inclinación que sugiere un comportamiento un poco más complejo. Esto sugiere que el tiempo de ejecución aumenta proporcionalmente con el tamaño de la entrada, lo que podría ser indicativo de una complejidad que ronda $O(n \cdot p \cdot m)$.

Requerimiento 8

Descripción

Breve descripción de cómo abordaron la implementación del requerimiento

Entrada	-Año de inicio de la consulta (con formato "%Y" ej.: “1989”) -Género de la película
Salidas	-Para el año y género de consulta informar: −El total de películas publicadas resultado de la consulta −El promedio de la votación promedio de estas películas de la consulta. −El tiempo promedio de duración de estas películas de la consulta −Las ganancias acumuladas por estas películas de la consulta −Nombre y puntuación de la película con mejor votación promedio publicada en el año. −Nombre y puntuación de la película con peor votación promedio publicada en el año.
Implementado (Sí/No)	Si - Grupal

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
current = year_list['first'] while current:	O(n)
for genero_prueba in genres_data:	O(m)
if genre in movie_genres and movie['status'] == 'Released':	O(1)
Total	O(n* m)

Pruebas Realizadas

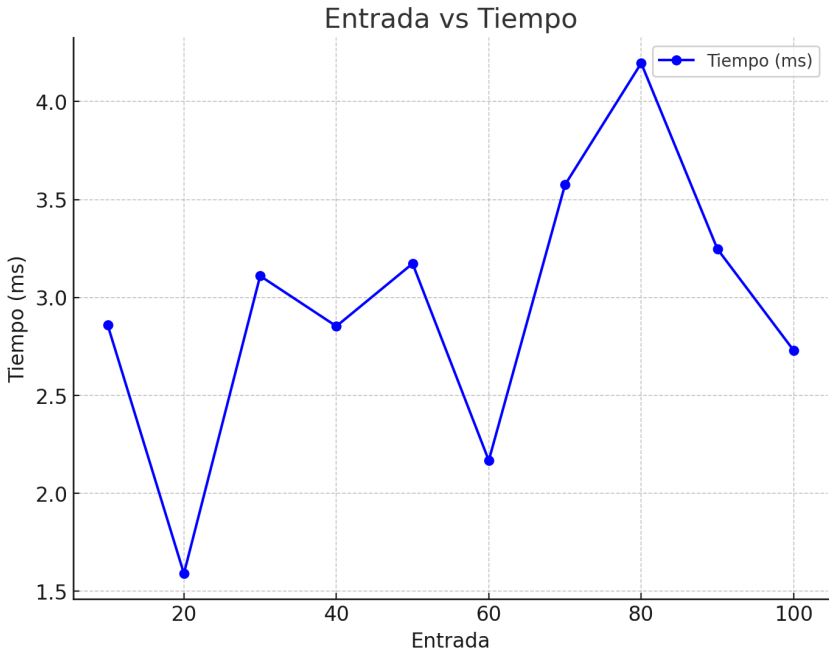
Las pruebas se realizaron en un computador con las siguientes características. La función recibió por parámetro los siguientes datos.

Máquina 3
AMD Ryzen 5 3550H
16,0 GB
Windows 11

Requerimiento 8 - Parametros: 1980 / Drama

Entrada	Tiempo
10	2.858999967575073
20	1.5907000303268433
30	3.110100030899048
40	2.8532999753952026
50	3.172999858856201
60	2.1679999828338623
70	3.5751999616622925
80	4.196200013160706
90	3.247099995613098
100	2.7297000885009766

Gráficas



Análisis

La gráfica muestra un comportamiento que es consistente con una complejidad

O(n · M), donde M no es una constante completamente fija. Las fluctuaciones que observamos en los tiempos de ejecución son una señal de que, aunque el tiempo de ejecución depende del número de elementos de entrada n, también existen otros factores internos (posiblemente relacionados con m que afectan el rendimiento en ciertos momentos.

Comparación Reto 1 / Reto 2

REQUERIMIENTO	PROMEDIO RETO 1 (aprox.)	PROMEDIO RETO 2 (aprox.)
Requerimiento 1	96.76 ms	0.1511 ms
Requerimiento 2	69.64	109294.71 ms.
Requerimiento 3	329.63 ms	107.51 ms
Requerimiento 4	394.56	4.17
Requerimiento 5	400.70	1135.44
Requerimiento 6	398.9	48.88
Requerimiento 7		139.64
Requerimiento 8	426.49	2.95

Conclusión: Como podemos percibir en la mayoría de funciones del reto 2 el tiempo de ejecución promedio fue menor comparado al del reto 1, exceptuando el segundo requerimiento y el quinto, pero en general se nota una mayor eficiencia.