

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC 3041 – Análisis y Diseño de Algoritmos

Sección 30

Julio Josh Mérida López



Notación Asintótica O y Máquinas de Turing

MICHELLE ANGEL DE MARIA MEJÍA VILLELA 22596

SYLVIA ALEJANDRA ILLESCAS FERNÁNDEZ 22376

DIEDERICH JOSUE EMIDIO SOLIS LOPEZ 22952

NIKOLAS DIMITRIO BADANI GASDAGLIS 20092

GUATEMALA, 7 de marzo de 2025

Notación Asintótica O :

Según los expertos, la notación Big-O es conocida por ser utilizada para analizar y describir el rendimiento de los algoritmos. Esto permite tener un mejor entendimiento acerca de la eficiencia temporal del algoritmo en el que respecta al crecimiento relativo del tamaño de la entrada que se le ingrese. Esta notación se centra en el peor de los casos y expresa el tiempo de la ejecución asintótica a medida que la entrada del algoritmo aumenta hasta llegar al infinito. La letra “O” que contiene ayuda a representar el orden de la magnitud. Mientras que los términos que se encuentran dentro del paréntesis ayudan a representar la función de complejidad del tiempo.

La base principal de la notación Big-O es la de proporcionar una forma sistemática y estandarizada de expresar la eficiencia temporal de un algoritmo en función del tamaño de la entrada ingresada. Al ser un lenguaje matemático, la notación permite ayuda a simplificar la complejidad del algoritmo, y a su vez, permite que los desarrolladores y análisis de datos puedan evaluar y comparar el rendimiento de los diferentes tipos de algoritmos.

Entre los conceptos clave más importantes para entender la notación Big-O se encuentran los siguientes : Tiempo Constante ($O(1)$), Tiempo logarítmico ($O(\log n)$), Tiempo lineal ($O(n)$), Tiempo cuadrático ($O(n^2)$) y Tiempo factorial ($O(n!)$). Estos ayudan para la realización del análisis de distintas formas :

- Tiempo constante : Indica que el tiempo de ejecución no varía conforme va aumentando el tamaño de los datos de entrada.
- Tiempo logarítmico : Indica que el algoritmo va creciendo en un principio, pero luego llega a un punto de estabilización conforme aumenta el tamaño de la entrada de los datos.
- Tiempo lineal : Indica que el tiempo de ejecución es proporcional a los datos de entrada.
- Tiempo cuadrático : Indica que el tiempo de ejecución del algoritmo aumenta de manera proporcional al cuadrado del tamaño de los datos de entrada.
- Tiempo factorial : Indica que, si el tamaño de la entrada aumenta, entonces el tiempo de ejecución va a aumentar de manera exponencialmente creciente.

Máquinas de Turing :

Concebida por el matemático Alan Turing en 1936, esta máquina es conocida por ser un modelo idealizado de la computación. Según lo indican los expertos, este modelo permite almacenar y procesar información virtualmente infinita. Su sistema operativo está compuesto por una abstracción matemática que se puede construir de un modo bastante sencillo, pero facilita la comprobación de un sinnúmero de preguntas relacionadas con teorías tanto de la computabilidad, como de la complejidad. Su nacimiento marco un gran momento en la historia de la informática. Y hasta el día de hoy, la máquina de Turing es considerada como el origen de las computadoras y ordenadores actuales.

Según lo indicado por los expertos, el tiempo de ejecución de la maquina se mide de acuerdo con la cantidad de pasos que realiza la maquina antes de detenerse por completo. Mediante el uso de la complejidad temporal, la máquina de Turing analiza los datos, y al final el resultado se expresa como una función del tamaño de la entrada n .

Entre los conceptos importantes para entender el funcionamiento de la maquina están los siguientes :

- Tiempo de ejecución : Se mide mediante el número de transiciones. Si la maquina no se detiene, entonces la ejecución es infinita.
- Complejidad temporal : Se expresa como una función $T(n)$. Para expresar el crecimiento del tiempo, se utiliza la notación Big-O.
- Clases de Complejidad :
 - o Tiempo polinómico
 - o Tiempo exponencial
 - o Tiempo no determinista polinómico
 - o Tiempo logarítmico

Convenciones de Entrada y Salida :

Entrada :

1. Numero entero (n) que es ingresado por el usuario
2. Archivo de configuración JSON
3. Selecccion en el menu principal para que el usuario pueda elegir entre dos opciones mediante la función input() :
 - a. Calcular un número de Fibonacci
 - b. Salir del programa

El usuario le proporciona al programa un numero entero (n) para poder calcular su término correspondiente en la secuencia de Fibonacci. La forma en la que el usuario puede ingresar el número es mediante la función input(). Para que, de esa forma, se convierta en a entero con la función int(). El archivo JSON es el que se encarga de los cálculos debido a que este contiene la configuración de la Maquina de Turing necesaria para realizar dichos cálculos. Sin mencionar que también ayuda a definir los estados, transiciones y acciones de la máquina.

Salida :

1. Traza de ejecución
2. Resultado final
3. Tiempo de ejecución
4. Mensajes de interacción
 - a. Inicio de la máquina
 - b. Ejecución del calculo
 - c. Opciones del menú
 - d. Mensajes de salida o errores

La traze de ejecución es la encargada de mostrar pasao a paso la evolución de la máquina de Turing. En la descripción de cada paso, se muestra el estado actual de la máquina, el contenido de la cinta y la posición del cabezal. Además, se imprimen en un formato que es legible para el usuario. Para cuando llega el momento del resultado final, este se muestra como valor de Fibonacci calculado para el numero n que fue ingresado por el usuario. El tiempo de ejecución que tomo la simulación generada por el programa se

muestra en segundos. Y por último, para ayudar al usuario a entender la estructura del programa y cómo funciona el programa, se muestran distintos mensajes de interacción.

Análisis Empírico :

El análisis se realizó con el objetivo de evaluar el desempeño de la maquina al ejecutar varias pruebas con diferentes valores de entrada (n). Los tiempos de ejecución de cada una de las pruebas se registran y se ajustan a un modelo matemático para así poder estimar su complejidad temporal.

Pasos del Análisis :

1. Medición del tiempo de ejecución
2. Generación de los gráficos
 - a. Gráfico de dispersión --> Tiempo de ejecución
 - b. Gráfico de regresión --> Tendencia de crecimiento
3. Cálculo del coeficiente de determinación (R^2)
 - a. Para evaluar la representación de los datos en el modelo matemático.

Casos de Prueba :

Prueba	Dato de Entrada (n)	Salida Esperada	Tiempo Esperado
Caso # 1	1	1	Tiempo mínimo
Caso # 2	5	5	Bajo
Caso # 3	10	55	Medio
Caso # 4	20	6765	Alto
Caso # 5	50	12,586,269,025	Muy alto

Para cada una de las pruebas, es importante verificar que el resultado de Fibonacci sea correcto, que el tiempo de ejecución siga una tendencia esperada, y por último, que los gráficos de dispersión y regresión reflejen la tendencia de crecimiento.

Finalmente, la interpretación de los resultados se puede expresar de la siguiente manera :

- Si los tiempos obtenidos en las pruebas demuestran estar siguiendo una curva exponencial o polinómica de alto grado, entonces la maquina posee una complejidad elevada.
- Si el valor de R^2 se encuentra cercano a 1, entonces el modelo de regresión si describe bien la relación que existe entre el dato de entrada (n) y el tiempo de ejecución.
- Si el valor de R^2 es bajo, entonces se puede deducir que hay una variabilidad alta en los tiempos de ejecución obtenidos, o que el modelo elegido no es el adecuado para interpretar los resultados.