Algoritmos sobre
Búsqueda y Ordenamiento
(Análisis y Diseño)

Dra. María de Luz Gasca Soto
Departamento de Matemáticas,
Facultad de Ciencias, UNAM.

17 de marzo de 2020

Luz Gasca S

Capítulo 7

La Cota Mínima del Ordenamiento

Los algoritmos de ordenamiento se pueden clasificar en dos grupos: los que emplean comparaciones para realizar el ordenamiento y los que no. Los que no emplean comparaciones tienen un mejor desempeño comparado con los que sí las utilizan, aunque tienen como desventaja cumplir ciertas suposiciones sobre la secuencia para poder ser usados.

Los métodos de ordenamiento que no emplean comparaciones son llamados técnicas de ordenamiento lineal ya que tienen un desempeño de $O(n)$. Por otro lado, el desempeño de las técnicas basadas en comparaciones oscila entre $O(n^2)$ y $O(n \log n)$. El desempeño de los métodos basados en comparaciones mejoró conforme se empleaban estrategias más ingeniosas o se usaban estructuras de datos auxiliares. Considerando esto, surgen las siguientes preguntas: ¿Será posible mejorar ese tiempo? ¿Cuál será el mejor desempeño para una técnica de este tipo?

Una Cota mínima para un problema particular es una evidencia de que *ningún algoritmo* puede resolver mejor el problema. Resulta más díficil probar una cota mínima, ya que tenemos que revisar *todos* los posibles algoritmos y no sólo uno. Necesitamos definir un modelo que corresponda a un algoritmo arbitrario y probar que el desempeño computacional para cualquier algoritmo que se ajuste a este modelo debe ser mayor o igual a la cota mínima.

Trabajaremos con un modelo de cómputo llamado Árbol de Decisión, el cual consiste principalmente de comparaciones. Los árboles de decisión no son modelos generales de computación como la máquina de Turing. Existen diversas variantes de los árboles de decisión, las cuales han sido utilizadas para obtener la mayoría de las cotas mínimas conocidas. Definimos árbol de decisión como árboles binarios con dos tipos de nodos: *nodos internos* y *hojas*. Cada nodo interno está asociado con una *pregunta* cuya respuesta es una de dos posibilidades, cada una de las cuales, a su vez, está asociada a una de sus ramas. Cada hoja está asociada a una posible salida o *pregunta*.

Así que, para responder las dos preguntas formuladas es conveniente abordar el problema con otro enfoque: en lugar de centrarnos propiamente en la técnica nos centraremos en el árbol de decisión asociado a ella. Informalmente, un árbol de decisión es un árbol binario en el cual cada hoja representa una posibilidad de orden. En la Figura 7.1 se presenta un árbol de decisión para tres elementos.

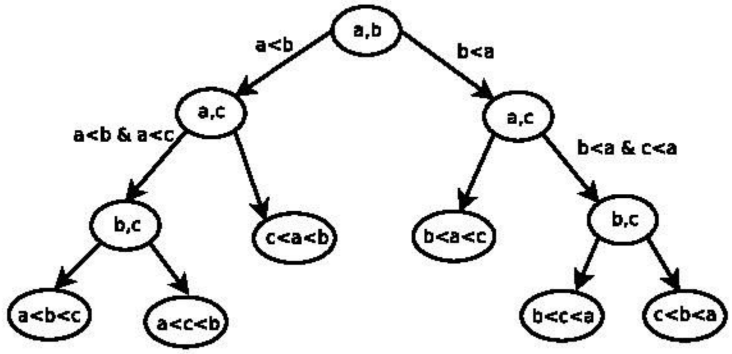

Figura 7.1: Árbol de decisión con tres elementos

Para este ejemplo suponemos que la entrada es una secuencia S de tres números: $S = \{a, b, c\}$. El cálculo empieza en la raíz del árbol. En cada nodo la pregunta se aplica a la entrada, de acuerdo a la respuesta se toma la rama izquierda o la derecha. En el ejemplo, se compara a a con b; si a resulta ser menor que b, toma la rama izquierda y ahora compara a a con c; si a es menor que c, entonces compara a b con c. Análogamente, se construye el sub-árbol derecho. De esta forma se va deduciendo el orden de la secuencia S. Cuando se llega a una hoja, la salida asociada a la hoja es justo el cálculo de comparaciones hechas para ordenar la secuencia S.

La acción de ordenar una entrada específica corresponde entonces a seguir un camino desde la raíz hasta una hoja, en donde los nodos internos por los que se pasa representan las comparaciones efectuadas por el algoritmo.

Es claro que, el tiempo de ejecución, en el peor de los casos asociado al árbol de decisión T, es la altura del árbol, la cual representa el máximo número de preguntas requeridas por la entrada. Un árbol de decisión construido de esta manera corresponde a un algoritmo. Aunque un árbol de decisión no puede modelar toda clase de algoritmos, son modelos razonables para algoritmos basados en comparaciones. Una cota mínima obtenida por árboles de decisión implica que *ningún* algoritmo basado en este modelo puede ejecutarse de mejor manera.

A continuación, usaremos los árboles de decisión para obtener la cota mínima de para el problema de ordenamiento. Revisaremos dos versiones que resultan ser equivalentes.

Primera Versión.

Sabemos que para una secuencia de n elementos, existen $n!$ maneras de acomodarlos, y como cada hoja dentro del árbol representa una posibilidad de orden, se infiere fácilmente que el árbol tendrá $n!$ hojas. Además, no necesariamente todas las hojas están al mismo nivel, como se aprecia en el árbol de la Figura 7.1.

Empleando un enfoque pesimista, tenemos que en el peor caso el número de comparaciones realizadas por una técnica de ordenamiento es igual a longitud de la trayectoria más larga desde la raíz hasta una hoja. Así, para obtener una cota mínima, en el peor caso, usando una técnica basada en comparaciones debemos determinar una cota inferior para la altura de un árbol, del cual sólo conocemos el número de hojas que tiene.

Como el árbol es binario entonces tiene a lo más 2^h hojas por lo que $n! \leq 2^h$, aplicando logaritmo base dos a ambos lados obtenemos: $\log(n!) \leq h$; es decir, la altura es, por lo menos, $\log(n!)$.

La expresión obtenida para la altura a simple vista se ve compleja, por ello hay que determinar cual es el valor de $n!$, si se emplea la aproximación de Stirling obtenemos $n! \geq (n/e)^n$ donde e es la base del logaritmo natural, entonces tenemos:

$$\log(n!) \geq \log((n/e\ )^n) = n(\log n - \log e) \approx\ n(\log n - 1{,}44{,}3n)$$

Es decir, en el peor caso se realizan $n \log n\ -\ 1{,}443n$ comparaciones. Por lo tanto, hemos encontrado la cota mínima de ordenamiento que es el mejor desempeño que se puede alcanzar, en el peor de los casos, con las técnicas de ordenamiento basadas en comparaciones y que redondeando es de $n \log n$.

Segunda Versión

Teorema 6.1 (Manber) Cada algoritmo tipo árbol de decisión para el problema de ordenamiento tiene altura $\Omega(n \log_2 n)$.

Demostración. Sea S la secuencia de números $S = \{x_1, x_2, ..., x_n\}$ el ejemplar a ordenar. Suponemos que la salida es una permutación de la entrada, la salida indica cómo reorganizar los elementos de tal manera que se transformen en una secuencia ordenada. Cada permutación es una posible salida, ya que la entrada puede estar en cualquier orden inicial.

Un algoritmo de ordenamiento es correcto si es capaz de reorganizar todas las posibles entradas en una secuencia ordenada. Así, cada permutación de $\{1, 2, ..., n\}$ deberá representar una posible salida en el árbol de decisión para ordenamiento. Cada salida, en el árbol, está asociada a una hoja. Dos diferentes permutaciones corresponden a dos diferentes salidas. Por lo tanto, debe haber, al menos, una hoja para cada permutación, es decir, para cada salida.

El número total de permutaciones para n elementos es $n!$, como el árbol es binario, su altura es de, al menos, $\log_2(n!)$. Ahora bien, de acuerdo con la fórmula de Stirling,

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + O(1/n))$$

De aquí, $\log_2(n!)$ es $\Omega(n \log_2 n)$, lo cual, completa la prueba. $\diamond$

Esta clase de cota es llamada una cota mínima teórica de información, ya que no depende los cálculos, sino de la cantidad de información contenida en la salida.

La cota mínima indica, en este caso, que cada algoritmo de ordenamiento requiere, en el peor de los casos, $\Omega(n \log_2 n)$ comparaciones, pues necesita distinguir entre los $n!$ diferentes casos y compara sólo dos posibilidades a la vez.

Si hubiésemos elegido un árbol de decisión donde cada nodo tuviese tres hijos, digamos, mayor, menor e igual, la altura del árbol sería, al menos, $\log_3(n!)$, la cual, de todas formas es $\Omega(n \log_2 n)$. Esto significa que la cota obtenida se aplica a cualquier árbol de decisión con un número constante de ramas por nodo.

La prueba del Teorema 6.1 implica que ningún algoritmo basado en comparaciones puede ser más rápido que $\Omega(n \log_2 n)$. Puede ser posible ordenar más rápido utilizando propiedades especiales sobre los datos o haciendo manipulaciones algebraicas sobre los datos.

Cabe señalar que, cuando discutimos sobre árboles de decisión, usualmente, ignoramos su tamaño y nos concentramos sólo en su altura. Un simple algoritmo de tiempo lineal puede corresponder a árboles de decisión con un número exponencial de nodos. El tamaño no es importante, ya que no intentamos construir el árbol, lo usamos únicamente como una herramienta para la demostración. Al ignorar el tamaño, hacemos más poderosa la prueba, ya que podemos aplicarla a programas de tamaño exponencial.

Por otro lado, esta técnica puede ser también poderosa. Los árboles de decisión son modelos de cómputo no uniformes. Dependen de n, el tamaño del ejemplar. Es potencialmente posible construir árboles para diferentes valores de n. También, es posible construir árboles de decisión de altura polinomial pero de tamaño exponencial, para problemas que probablemente requieran tiempo de ejecución exponencial.

Tenemos que los árboles de decisión son algunas veces optimistas. Esto es, un árbol de decisión para la cota mínima puede estar muy por debajo de la complejidad real del problema. Por otro lado, si la cota mínima es igual a la cota máxima para un algoritmo en particular, entonces la cota mínima implica que aún si usáramos más espacio, no es posible mejorar el desempeño del algoritmo.

Cabe mencionar que el caso promedio de cualquier algoritmo basado en comparaciones para el problema de ordenamiento también es $\Omega(n \log_2 n)$.