Introducción

No busco insinuar que he descubierto este algoritmo que llamo ordenamiento HRH; tan sólo deseo presentar el breve y escueto análisis que he podido realizar con lo poco que sé de algoritmia. Por varias semanas estuve preguntándome si valía la pena publicar los resultados; nada me cuesta reconocer que me entusiasma un poco compartir esa simplicidad que me llevó a intentar entender el algoritmo.

El ordenamiento HRH es bastante similar al ordenamiento por selección, con la diferencia de que el primero identifica simultáneamente el menor y el mayor elemento en cada iteración, llevándolos a los extremos del arreglo. Quizás el siguiente pseudocódigo ayude a comprenderlo un poco mejor:

```
hrh_sort(A):
{
    a = 0
    b = n - 1
    Mientras a <= b
    {
        minimo = a
        maximo = a
        Para i = a hasta i = b
        {
            Si A[i] < A[minimo]:
                minimo = i
            Si A[i] > A[maximo]:
                maximo = i
        }
        intercambia(A[a], A[minimo])
        Si maximo == a:
            maximo = minimo
        intercambia(A[b], A[maximo])
        a = a + 1
        b = b - 1
    }
}
```

Para determinar el número de comparaciones realizadas para identificar los valores mínimo y máximo, debemos notar que en cada iteración del bucle principal se evalúan todos los elementos del subarreglo $A[a:b]$ actual, reduciendo su tamaño n en dos elementos por iteración (a avanza hacia la derecha y b hacia la izquierda). Esto da lugar a una de dos progresiones aritméticas decrecientes.

En el caso de n par, el número de comparaciones S_1 puede representarse de la siguiente manera:

$$S_1 = n + (n-2) + (n-4) + ... + 6 + 4 + 2.$$

En esta progresión, la fórmula del k-ésimo término es:

$$a_k = n - 2(k-1).$$

Buscamos encontrar el número de términos k sumados, o lo que es lo mismo, el número de iteraciones completas del bucle cuando $a_k = 2$. Tras el despeje, obtenemos que $k = \frac{n}{2}$.

Recordemos que la suma de una progresión aritmética se calcula con la fórmula:

$$S = \frac{k(a_0 + a_k)}{2}.$$

Sustituyendo:

$$S_1 = \frac{\frac{n}{2}(n+2)}{2} = \frac{n(n+2)}{4}.$$

En el caso de n impar, la progresión es bastante parecida, pero en este caso el último término será el número 1:

$$S_2 = n + (n-2) + (n-4) + ... + 1.$$

La fórmula para el k-ésimo elemento sigue siendo:

$$a_k = n - 2(k-1),$$

aunque ahora consideraremos $a_k = 1$, de donde se sigue que $k = \frac{n+1}{2}$ y que la progresión aritmética equivale a:

$$S_2 = \frac{(n+1)^2}{4}.$$

Así, el número total de comparaciones realizadas por el algoritmo HRH puede definirse con la siguiente función $h(n)$:

$$h(n) = \begin{cases} \frac{n(n+2)}{4} & \text{si } n \bmod 2 = 0 \\ \frac{(n+1)^2}{4} & \text{si } n \bmod 2 \neq 0 \end{cases}$$

Por último, con base en el análisis anterior, determinaremos su complejidad temporal.

Notemos que ambos casos de $h(n)$ tienen un término dominante $\frac{n^2}{4}$. Para verificar formalmente la relación asintótica, tomaremos $f(n) = \frac{n^2}{4}$ y $g(n) = n^2$ como referencia. A continuación evaluaremos:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)},$$

que equivale a:

$$\lim_{n \to \infty} \frac{n^2}{4n^2} = \frac{1}{4}.$$

Ya que el resultado del límite es un número real finito, concluimos que $h(n) \in \Theta(n^2)$.

Bajo el modelo analizado, el HRH reduce las comparaciones realizadas por el ordenamiento por selección estándar, aunque ambos comparten la misma complejidad temporal cuadrática; además, la diferencia práctica entre los algoritmos es mínima, lo que probablemente se deba a cómo se realizan las comparaciones.

Eso es todo. Quizás haya algo valioso en estas páginas o en el algoritmo; quizás pueda encontrarse en ellos el tímido asombro con que hace algunos meses comencé a trabajar en esto, acaso valioso por su lentitud, acaso valioso por su brevedad.

V. L. V.
Zapopan, 23 de diciembre de 2024