

# Historias de la Historia de la Computación

## Veintiún Líneas de Código que Cambiaron la Historia

Felipe Ovares Barquero

felipe\_ovares@yahoo.com

Universidad Nacional, Escuela de Informática



Volumen 7, Número 4

### I- Verano rojo en Moscú

Ahora, Sir Charles Antony Richard Hoare tiene un aspecto simpático y decimonónico. En algunas fotografías me recuerda a cierto mago que maravillaba a la clientela con una baraja, sin embargo, es un científico británico en Computación. Me lo imagino, con veinticinco años, vacacionando en una rústica cabaña ubicada en un promontorio desde donde se ve el lago Glubokoe. En julio este es un paraje apacible. En la mesa de madera del corredor, nuestro héroe, está jugando a las cartas con sus amigos. Como en casi todos los juegos de baraja, es necesario ordenar las cartas de prisa para intentar ganar. Está obsesionado con el ordenamiento, no de las cartas, ni de los números, necesita un algoritmo para ordenar un diccionario y poder efectuar búsquedas rápidas de palabras para un traductor [3]. Acaba de recoger sus cartas de la mesa, las observa, ahora debe ordenarlas. El dos de corazones, para la escalera, debe ir hacia la izquierda, al lado del cuatro de oros, en seguida acomoda el cinco de bastos, no le ve futuro a la jugada, pero no tiene alternativa. Espera, el azar es caprichoso. La próxima carta también tendrá que colocarla en orden. Las preguntas que lo inquietan, desde algún tiempo, continúan siendo las mismas: ¿Cómo podemos ordenar las cartas tan fácilmente? ¿Cómo se lo explico a una computadora? ¿Cómo hacerlo con la menor cantidad de movimientos?

Tony, como lo llamaban sus camaradas soviéticos, estudió filosofía en la Universidad de Oxford, que en aquél tiempo exigía aprender latín y griego, se enamoró de la lógica y luego de la computación. Mientras hacía el servicio militar en la Royal Navy aprendió ruso, circunstancias o azar no sé, ni quiero aventurarme con la especulación. Agregaré que accidentalmente, se matriculó en un curso de programación Fortran con Leslie Fox, notable profesor, que fue además durante varios años, funcionario del servicio secreto de su Majestad. En 1959, en plena guerra fría, Tony se encuentra en Rusia estudiando en la Universidad de Moscú, en la Escuela de Andrey Nikolayevich Kolmogorov, traducción automática y teoría de probabilidades. Es en esa misma época que me lo imagino en el lago pensando cómo ganar el juego, pero, sobre todo, atribulado con la obsesión de hallar una forma para ordenar un conjunto de palabras que facilite una búsqueda rápida. Ya sabe que si el conjunto está ordenado la búsqueda requiere  $\log_2 N$  intentos, donde  $N$  es la cantidad de elementos del conjunto, pero cada vez que ingresa un nuevo elemento tiene que volver a ordenarlo y eso es ineficiente y los algoritmos que conoce para ejecutar el procedimiento no cumplen con sus expectativas, debe haber algo mejor. Si el conjunto no está ordenado la cantidad de intentos antes de localizar el elemento es muy alta y eso no le gusta porque su traductor sería demasiado lento.

Vuelve a recoger las cartas. Ha perdido cuarenta y siete rublos con veintitrés copeks. Si estuviera jugando al veintiuno estaría ganando, su abuela le enseñó desde pequeño los trucos del conteo. Pero con el Ron de Contrato, versión rusa, el asunto es diferente y además no está concentrado. Selecciona el 6 de trébol, lo coloca en el centro de sus cartas, pasa a la izquierda los menores y a la derecha los mayores, ahora está seguro de que el

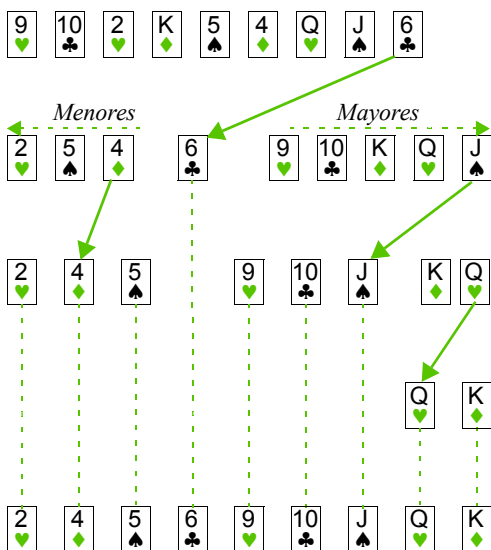


Figura 1. Así ordenó las Cartas.

6 no se moverá de ahí, repite el procedimiento con los dos grupos que lo rodean, en el de la derecha toma la J de bastos, luego coloca los menores a la izquierda y los mayores a la derecha, con el grupo de la izquierda resulta fácil el ejercicio, toma el cuatro deoros colocándolo en el centro de las tres cartas y por casualidad el grupo queda ordenado. Vuelve a hacer lo mismo hasta que los grupos terminan ordenados (Figura 1). Es una solución típica de “divide y vencerás”. Lo mejor es que funciona... Insiste en las siguientes jugadas. Saca una factura de la billetera que le dieron la semana anterior en un restaurante cercano a la calle Tverskaya en el centro de Moscú, por el lado de atrás garabatea un ejemplo.

A la tres de la mañana no resiste la tentación y se levanta de la cama donde no ha podido conciliar el sueño, regresa al corredor, allí mediante la complicidad de la luz de una linterna del PKKA escribe en pseudo código las sentencias para ordenar un vector que se le ocurrieron durante el juego. Al día siguiente, en su apartamento lo traduce a Fortran. Lo prueba varias veces en el cuaderno y funciona. En la universidad lo pasa a las tarjetas y lo ejecuta hasta que el programa se comporta como él desea: ordenando un vector con 15 números enteros. Tony Hoare ha inventado el **Quicksort** [2, 4]. La versión recursiva la programó, meses después, en Algol 60. Dijkstra hace el siguiente comentario en 1999: “Para darles una idea de lo maravilloso de esta sentencia –se refiere a la recursividad– les diré que fue crucial al permitirle a Tony Hoare completar el diseño de uno de los algoritmos más famosos: el

```

1. QuickSort(lista, inf, sup)
// Inicialización de variables
2. pivote = lista[sup];
3. i = inf;
4. j = sup - 1;
5. cont = 1;
// Verifica que no se crucen los límites
6. if (inf >= sup)
7.     retornar;
// Clasifica la sublista
8. while (cont)
9.     while (lista[++i] < pivote);
10.    while (lista[--j] > pivote);
11.    if (lista[i] > lista[j])
12.        temp = lista[i];
13.        lista[i] = lista[j];
14.        lista[j] = temp;
15.    else
16.        cont = 0;
// Copia el pivote en su posición final
17. temp = lista[i];
18. lista[i] = lista[sup];
19. lista[sup] = temp;
// Aplica el procedimiento recursivamente a cada sublista
20. QuickSort (lista, inf, i - 1);
21. QuickSort (lista, i + 1, sup);
    
```

Figura 2. Las 21 líneas del Quicksort.

*quicksort. Antes de que él conociera el Algol, tenía la idea, pero permaneció en una forma evasiva, él observó que el Algol 60 le ofrecía lo que necesitaba y el Quicksort se materializó en toda su gloria” [1].*

El **Quicksort** es el algoritmo de ordenamiento más popular en el mundo y es, además, el algoritmo de cualquier tipo más utilizado. En todas las pruebas conocidas supera a los demás.

## 2- Invierno gris en Londres

Año 2000. Soy un turista desinformado en las inmediaciones del Palacio de Buckingham. Una pareja vestida para alguna ocasión festiva posa para los fotógrafos de la prensa. Laura, mi amiga inglesa, no los conoce, pero estamos tan cerca que podemos apreciar la felicidad de ambos y una condecoración que según ella solo se la otorgan a los caballeros. En otras palabras, la ceremonia que recién ha terminado en el palacio fue para premiar al hombre de la fotografía (Figura 3). Como sigo intrigado con la situación Laura se despista un momento y regresa con la respuesta: “se llama Charles Antony Richard Hoare. Lo premiaron por su contribución en el campo de la computación y sus servicios a la educación. Ahora es Sir Charles y usted debería conocerlo”. Por supuesto, me sonrojo, imparto el curso de Estructuras de datos... Cuando reacciono luego de la sorpresa el grupo de fotógrafos y curiosos se ha desordenado y ya Tony y su esposa Jill están ingresando en un emblemático taxi londinense. Me hubiera encantado una fotografía con ellos.

El 27 de octubre de 1980, en Nashville, Tennessee, Tony Hoare había sido galardonado con el premio Alan Turing de la ACM (Association for Computer Machinery), considerado el Nóbel de Computación, por sus contribuciones fundamentales a la definición y diseño de lenguajes de programación.

Tony nació el 11 de enero de 1934 en Colombo, Sri Lanka y aún no se ha pensionado, ahora trabaja en investigación para Microsoft en Cambridge Inglaterra. Tony Hoare también es creador del lenguaje formal CSP (tercer trabajo más citado en computación y diseño de lenguajes de programación).

Tony resume así su experiencia con el algoritmo: “*Mi historia se inicia en agosto de 1960, cuando comencé a programar en una pequeña empresa de manufactura, una división de Elliott Brothers, mi primera tarea fue implementar, para la nueva computadora Elliott 803, una biblioteca de subrutinas para el nuevo método de ordenamiento interno inventado por Donald Shell [5]. Mi jefe y tutor Pat Shackleton estaba muy contento con mi programa. Entonces le dije tímidamente que yo había inventado un método de ordenamiento más rápido que el Shell, sin requerir demasiado almacenamiento adicional. Él apostó seis peniques a que no. Aunque mi método era difícil de explicar al final aceptó y yo gané la apuesta*”.

### 3- Consideraciones Esenciales

La selección del pivote (Figura 2) afecta considerablemente el tiempo de ejecución del algoritmo, del caso óptimo al peor varía de  $O(n \log_2 n)$  a  $O(n^2)$ . El pivote ideal divide la lista en partes iguales. El peor caso es cuando el vector está ordenado porque cada llamada genera una sublista y no dos. Este problema es frecuente cuando se agrega un valor a un vector ordenado. El algoritmo, al depender de la suerte de un buen pivote, no es estable. Para mejorarlo se recomienda tomar el primer elemento del vector, el que está en medio y el último, luego se comparan los tres valores y se escoge el que está en medio.

**Ejercicio.** Para un vector con 10 elementos el algoritmo de la *Burbuja* es más rápido que el Quicksort, pero para cantidades más grandes sucede lo contrario. Investigar a partir de que cantidad el **Quicksort** es más rápido que la *Burbuja*. ¿Tiene alguna importancia conocer este dato?



Figura 3. Tony y Jill en Londres.

### Referencias

- [1] E. Dijkstra, “*Computing Science: Achievements and Challenges*”, Keynote address ACM Symp. on Applied Computing at San Antonio, TX, March 1999.
- [2] C. A. R. Hoare, “*Algorithm 64: Quicksort*”. Commun. ACM 4(7): 321-322, Julio 1961.
- [3] C. A. R. Hoare, “*A method of Synthesizing Sentences in Machine Translation on the Basis of Syntagmatic Analysis*”, Foreign Develop. Mach. Translat. Info. Proc. No. 95 (Translated from Mashinnii Pervod i Prikladnaya Linguistika No. 6), 1961
- [4] C. A. R. Hoare, “*Quicksort*”, Computer Journal, 5(1):10-15, 1962.
- [5] D. L. Shell, “*A high-speed sorting procedure*”, Communications of the ACM 2 (7): 30-32, 1959.

*Felipe Ovares Barquero es profesor en la Universidad Nacional de Costa Rica en donde imparte los cursos de Sistemas Operativos y Estructuras de Datos para la Escuela de Informática. También es el Director-Editor de la revista Uniciencia de la Facultad de Ciencias Exactas y Naturales. Obtuvo una licenciatura en Computación en la Universidad de Costa Rica. Recién publicó un libro titulado “Cien mentiras y otros cuentos”. La mayoría de sus relatos se pueden leer en [www.felipeovares.com](http://www.felipeovares.com)*

