¿Quién fue Edsger W. Dijkstra?

Dijkstra nació en 1930 en Róterdam, Países Bajos. Su padre fue un profesor de preparatoria de química y su madre siempre fue una gran influencia para él entorno a las matemáticas y la elegancia. Al terminar sus estudios de preparatoria, intentó ser un físico teórico. Sin embargo, Dijkstra pensó que la habilidad de usar una computadora electrónica podría traerle grandes ventajas. Tres años programando en el Centro Matemático en Ámsterdam lo convenció de que el reto intelectual de la programación superaba al requerido por la física teórica. Su jefe, A. van Wijngaarden, los persuadió de que en los años venideros, él fuera una de las personas que hiciera de la programación una disciplina respetable.

Cuando trabaja en el Centro Matemático, él desarrolló dos de los más conocidos algoritmos en teoría de grafos, es decir, el algoritmo de búsqueda del camino más corto y el del árbol de expansión mínima como parte de un proyecto de desarrollo de computadoras. Ambos algoritmos fueron publicados en un solo artículo científico. Durante su tesis doctoral, se concentró en el problema de un interruptor en tiempo real. Junto con J. A. Zonneveld desarrolló el primer compilador para Algol-60. Él fue el primero en implementar recusión y el uso de pilas para traducir programas recursivos. En el diccionario de inglés de Oxford, los términos "vector" y "stack" en el contexto computacional son atribuidos a Dijkstra. Él también diseñó un sistema operativo para una computadora que se desarrollaba en le Universidad Técnica de Eindhoven. Dijkstra publicó un artículo breve en la revista "Communications of ACM" acerca de que la sentencia "go to", presente en muchos lenguajes de programación de alto nivel, debería ser eliminada pues era la fuente de muchos y graves errores en programas de cómputo. El principio de su camino hacia el premio Turing fue durante la conferencia "1968 NATO Conference on Software Engineering" donde se dio cuenta de la inmensidad de la llamada crisis del software. Es ahí donde diseñó sus primeras ideas acerca de formular a la programación como parte de una disciplina matemática. Aquí, él hizo énfasis en que la productividad y veracidad o correctitud de los programas era similar al rigor en matemáticas. Por este tipo de ideas, en 1972, Dijkstra fue galardonado con el premio Turing.

Cabe destacar que la carrera científica de Dijkstra fue altamente productiva. Él se retiró de la enseñanza en Noviembre de 1999, después de más de 40 años de contribuciones a las ciencias en computación. Regresó a Países Bajos en Febrero de 2002 y en Agosto de ese año, murió.

¿Por qué recibió el premio Turing?

Dijkstra fue galardonado con el premio más importante en computación debido a sus contribuciones fundamentales en hacer de la programación un alto reto intelectual; por su elocuente insistencia y demostración práctica acerca de que los programas deben ser compuestos correctamente y no sólo corregidos hasta la correctitud. Además, por destacar los problemas fundamentales acerca del fundamento del diseño de programas.

¿Por qué elegí a Dijkstra?

Elegí a este gran personaje de la computación debido a su filosofía de diseñar programas correctamente. Dado que actualmente las ambiciones de los programas de cómputo son

cada vez mayores, no es posible entrar en un ciclo infinito de correcciones al programa por el hecho de no haberlo diseñado correctamente. Asimismo, los programas con una expresión de las matemáticas, por lo que si el programa no funciona correctamente, entonces el fundamento matemático detrás está equivocado. En mi investigación me he dado cuenta de que primero pensar y después programar lleva a ahorrar muchas horas que se desperdiciarían buscando errores en los programas.

Sobre lo que me motiva

Hay dos motivaciones principales que me dejaron las palabras de Dijkstra. Primeramente, al momento de realizar un programa siempre buscar realizarlo de manera correcta y completa desde la primera ocasión. Esto con el objeto de evitar liberar versiones con correcciones o "parches". Siempre debe buscarse un producto completamente terminado. En segunda instancia, es muy cierto que cuando uno de nuestros programas no funciona adecuadamente no podemos decir que es algún problema sobrenatural en la computadora. Un error en un programa ocurre debido a un error en su concepción. Como bien dice Dijkstra, en el caso de los programas su fallo no se le puede atribuir a la obstinada naturaleza.

Sobre lo que me gusta

Buscar crear programas con calidad, correctitud y elegancia. Es una filosofía que claramente no sólo la gente relacionada con ciencias de la computación debe adoptar sino que, en general, todos debemos seguir. El segundo aspecto que me agradó fue la comparación entre Beethoven y Mozart. En este caso, me agrada la idea de sólo escribir el programa una vez haya sido completamente formulado y no ir programando conforme se va diseñando la función a ejecutar. Esto claramente va de la mano con filosofía que menciona Dijkstra. Finalmente, me gustó bastante la manera de crear sus manuscritos. A mí agrada llevar esa misma metodología.

Sobre lo que me sorprende

Es sorprendente cómo su plática en Bruselas fue relativamente un fracaso debido a que la compañía de software no deseaba un producto sin fallos. Claramente, se antepone la parte económica a tener un producto de calidad. Finalmente, lo que más me sorprendió fue el caso del Apollo. Sin duda alguna, la correctitud de los programas es vital en aplicaciones críticas ya que algún de error puede causar consecuencias desastrosas.