

Edsger W. Dijkstra

Life and Death

Edsger Wybe Dijkstra (11th May 1930 – 6th Aug 2002) was born in Rotterdam. Dijkstra initially considered a career in Law however after completing school at the age of 18, took his parents suggestion to study mathematics. He obtained a M.S in Mathematics and Physics from the University of Leiden in 1956 before completing a PhD in Computing Science from the Municipal University of Amsterdam three years later.

In the 1950's, computers were a rare commodity and so Dijkstra seemed to stumble upon his path of software engineering quite accidentally. Through a referral from his supervisor, Dijkstra was introduced to the director of the Computation Department at the Mathematical Centre in Amsterdam who subsequently offered him a job, making Dijkstra the Netherlands first official "programmer". (1) As he said himself in his paper titled "The Humble Programmer"

"As a result of a long sequence of coincidences I entered the programming profession officially on the first spring morning of 1952 and as far as I have been able to trace, I was the first Dutchman to do so in my country." (2)

5 years later, when Dijkstra married Maria C. Debets in 1957, he was required as part of the marriage rites to state his profession. Dijkstra stated he was a programmer, however this was unaccepted by the authorities as it was deemed an unofficial profession and as a result Dijkstra had to settle for the profession of a theoretical physicist! (2)

Dijkstra worked in the Mathematical Centre in Amsterdam from 1952 to 1962. His main role was to write the software for the at the then time non-existent computer, which was being designed by colleagues Bram Jan Loopstra and Carel S. Scholten.

In 1962, Dijkstra moved to the south of the Netherlands, where he became a professor for the Mathematical Department at the Eindhoven University of Technology. He had not originally been the university's first preference for the position and was in-fact their third, due to the fact he had not previously really studied mathematics and for other reasons too.

"the decision to invite me had not been an easy one, on the one hand because I had not really studied mathematics, and on the other hand because of my sandals, my beard and my "arrogance" (whatever that may be)". (3)

Dijkstra then joined Burroughs Corporation as it's Research Fellow in 1973. His duties consisted of visiting some of the company's research centres a few times a year and carrying on his own research, which he did in the smallest Burroughs research facility, namely, his study on the second floor of his house in Nuenen. Dijkstra accepted the Schlumberger Centennial Chair in the Computer Science Department at the University of Texas at Austin in 1984. It was here that Dijkstra worked until his retirement in November 1999. (1)

Dijkstra and his wife returned from Austin to Amsterdam where he found out he was terminally ill. Dijkstra died on 6th August 2002 after a long struggle with cancer. He and his wife were survived by their three children: Marcus, Femke and Rutger M. Dijkstra (who also became a computer programmer).

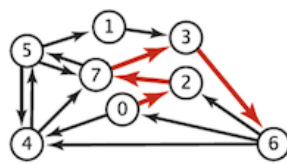
Work and Contribution

During Dijkstra's time in the Mathematical Centre in Amsterdam, was when he began his work on the famous shortest path problem. In graph theory, the shortest path problem is the problem of finding the shortest possible path between two nodes in a graph such that the sum of the weights of its constituent edges is minimized.

Dijkstra formulated and solved the Shortest Path problem for a demonstration at the official inauguration of the ARMAC computer (one of the first computers built in the Netherlands) in 1956, but because of the absence of journals dedicated to automatic computing he did not publish the result until 1959. The algorithm was still not generally known for several years after that. (1)

edge-weighted digraph

4→5	0.35
5→4	0.35
4→7	0.37
5→7	0.28
7→5	0.28
5→1	0.32
0→4	0.38
0→2	0.26
7→3	0.39
1→3	0.29
2→7	0.34
6→2	0.40
3→6	0.52
6→0	0.58
6→4	0.93



shortest path from 0 to 6

0→2	0.26
2→7	0.34
7→3	0.39
3→6	0.52

Example retrieved from the Oxford Math Centre

The Algorithm

1. Assign to every vertex a value which will store an associated distance. For the source vertex, make this distance zero. For every other vertex, initially make this distance infinite.
2. Mark all nodes as unvisited.
3. Set the "current vertex" initially to be the source vertex.
4. Repeat the following until all vertices have been visited:
 - For the current vertex, consider all its unvisited neighbours and calculate their tentative distance (i.e., the sum of the distance associated with the current vertex plus the distance between the current vertex and the unvisited neighbour in question).
 - If this distance is less than the previously recorded distance for the current vertex, overwrite the current vertex distance with this smaller distance (edge relaxation) and mark the current vertex as visited.
 - Set the unvisited node with the smallest distance from the source node as the next "current node"

Dijkstra also invented an algorithm to calculate the minimum spanning tree of a network. The minimum spanning tree of a network is defined as the shortest path between all edges to link every node in the network. The asymptotic running time of the algorithm grew as a square of the network size, which was extremely efficient for the 1950's. This algorithm along with the shortest path, were published in a 3 page article "A note on two problems in connexion with graphs" in 1959. (4)

Minimum Spanning Tree Algorithm

The edges are subdivided into three sets

- i. The branches definitely assigned to the tree under construction (they will form a subtree)
- ii. The branches from which the next branch to be added to set I, will be selected;
- iii. The remaining branches (rejected or not yet considered)

The nodes are subdivided into two sets:

- A. The nodes connected by the branches of set I
- B. The remaining nodes (one and only one branch of set II will lead to each of these nodes)

We start the construction by choosing an arbitrary node as the only member of set A, and by placing all branches that end in this node in set II. To start with, set I is empty. From then onwards we Perform the following two steps repeatedly.

1. The shortest branch of set II is removed from this set and added to set I. As a result one node is transferred from set B to set I.
2. Consider the branches leading from the node, that has just been transferred to set A, to the nodes that are still in set B. If the branch under consideration is longer than the corresponding branch in set II, it is rejected; if it is shorter, it replaces the corresponding branch in set II, and the latter is rejected.

Dijkstra also made a substantial contribution to the world of recursive programming, as he was the first to introduce the concept of the "stack" used in recursive algorithms which were subsequently published in "Numerische Mathematik", the following year. Dijkstra described the use of the stack – "One uses a stack for storing a sequence of information units that increases or decreases at one end only, i.e. when the unit of information that is no longer of interest is removed from the stack then this is always the most recently added unit still present in the stack." (5)

Impact and Recognition

There is no denying that Dijkstra paved the way for discovering fundamental aspects of computer programming as well as inventing many algorithms and methods himself. Although the above mentioned algorithms and concepts only represent a fragment of his life's work, they are without a doubt the most influential.

In today's society, the shortest path problem and minimum spanning tree problem are both widely discovered and solved every day. For example the shortest path algorithm is used to calculate travel distance or time between two places on a map. The minimum spanning tree can be simplified into a delivery man having to deliver to several different warehouses in different locations and how to do so in the shortest distance or time. The problems which Dijkstra had thought so much about, are still thought about every single day.

Dijkstra's Implementation of the stack used in recursive programming can be seen in everyday life. For example, undo and redo in Microsoft office is based on the stack principle as well as many other basic algorithms such as stock taking methods FIFO and LIFO. In addition to Dijkstra's algorithmic work, it appears the biggest contribution he made to the world of software engineering was his powerful work ethic and dedication.

The ACM Turing Award is an annual prize given by the Association for Computing Machinery (ACM) to an individual selected for contributions of lasting and major technical importance to the computer field. In 1972, Dijkstra received this award and the introductory speech from the award ceremony outlines the impact he made to the computing world. (6)

"The working vocabulary of programmers is studded with words originated or forcefully promulgated by E. W. Dijkstra: display, deadly embrace, semaphore, go-to-less programming, structured programming. But his influence on programming is more pervasive than any glossary can possibly indicate. The precious gift that this Turing Award acknowledges is Dijkstra's style: his approach to programming as a high, intellectual challenge; his eloquent insistence and practical demonstration that programs should be composed correctly, not just debugged into correctness; and his illuminating perception of problems at the foundations of program design."

Other notable honours and awards include:

- Member of the Royal Netherlands Academy of Arts and Sciences (1971)
- Distinguished Fellow of the British Computer Society (1971)
- AFIPS Harry Goode Memorial Award (1974)
- Foreign Honorary member of the American Academy of Arts and Sciences (1975)
- Doctor of Science Honoris Causa from the Queen's University of Belfast (1976)
- Computer Pioneer Award from the IEEE Computer Society (1982)
- ACM/SIGCSE Award for outstanding contributions to computer science education (1989)
- ACM Fellow (1994)
- Honorary doctorate from the Athens University, Greece (2001)
- ACM Influential Paper Award for his paper, "Self-stabilizing systems in spite of distributed control" (2002) (7)

Conclusion

Edsger W. Dijkstra discovered and introduced several algorithms and concepts which remain widely used today. He was the pioneer of software engineering, undoubtedly paving the way for software engineers after him. His contribution to computing as well as his hard-working attitude saw him a worthy winner of the ACM Turing award.

Bibliography

- (1) Faulkner, Larry R; Durbin, John. R (25 August 2003). “In Memoriam: Edsger Wybe Dijkstra”. The University of Texas at Austin.
- (2) Edsger W. Dijkstra. The humble programmer. Communications of the ACM 15, 10 (1972), 859–866. Turing Award lecture.
- (3) Edsger W. Dijkstra. From my life, EWD 1166.
- (4) Edsger W. Dijkstra. A note on two problems in connection with graphs. Numerische Mathematik 1 (1959), 83–89.
- (5) Edsger W. Dijkstra. Recursive programming. Numerische Mathematik 2 (1960), 312–318.
- (6) Extract from the Turing award Citation ready by M.D. McIlroy, chairman of the ACM Turing Award Committee, 14, 1972, at the ACM Annual Conference in Boston. Available: https://awards.acm.org/award_winners/dijkstra_1053701.cfm [Accessed 28th Oct. 2018]
- (7) Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. Communications of the ACM 17, 11 (1974), 643–644.