

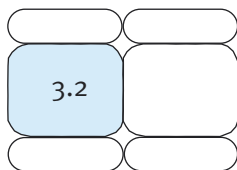


E.W. DIJKSTRA

UNA VITA DA INFORMATICO

Lorenzo Milone

Edsger Wybe Dijkstra è stato un informatico poliedrico e dai numerosi interessi, un pioniere che ha attraversato tutte le fasi dell'informatica moderna, dalla nascita alla maturità. L'articolo ne ripercorre l'intera vita, personale e professionale mettendone in evidenza i periodi fondamentali.



1. INTRODUZIONE

Dijkstra. Avevo ascoltato più volte quel nome strano, pronunciato un po' come veniva¹. Ricordo che stavo studiando gli algoritmi. Ne avevo visti tanti ma uno tra tutti mi aveva particolarmente colpito per eleganza e semplicità: era l'algoritmo dei cammini minimi in un grafo pesato ed era associato a quel nome. Poi è arrivato il tempo dello studio dei sistemi operativi. Problemi complicati e, di conseguenza, soluzioni complesse, eccetto quelle associate a quel nome. Geniale, ad esempio, la soluzione al problema della mutua esclusione: i semafori. A questo punto la curiosità è troppo forte. È giunto il momento di provare a rispondere ad una domanda che spesso, in questi anni, mi sono posto: chi era Dijkstra?

Edsger Wybe Dijkstra è stato un informatico poliedrico e dai numerosi interessi. Lo scopo di questo contributo è quello di ripercorrerne, seppure in maniera preliminare, l'intera vita, personale e professionale, e di metterne in evidenza i periodi fondamentali. Ad ogni periodo

sarà dedicato un paragrafo del presente lavoro. Un paragrafo a parte sarà dedicato ai suoi manoscritti, anche noti come la serie EWD.

Nel secondo paragrafo analizzeremo gli anni giovanili a Rotterdam. Nel terzo, i primi approcci con l'informatica e il lavoro al Centro Matematico di Amsterdam. Nel quarto, l'insegnamento all'Università di Eindhoven. Nel quinto, la collaborazione con la Burroughs Corporation. Nel sesto, il periodo all'Università di Austin in Texas. Nel settimo, esamineremo i manoscritti, inesauribile e stimolante fonte primaria sulla vita e sull'attività professionale di Dijkstra.

2. I PRIMI ANNI

Edsger Wybe Dijkstra nacque a Rotterdam l'11 maggio 1930, terzo di quattro figli. Sua madre, Brechtje Cornelia Kluyver, era un matematico²; suo padre, Donwe Wybe, un chimico.

Dal 1942 al 1948 frequentò il Gymnasium Erasmusianum a Rotterdam. Durante l'ultimo an-

¹ "Dijkstra", "Daikstra". La pronuncia corretta, letta in italiano, è "Deikstra": "Etscher Uibe Deikstra".

² Dijkstra ricorda che la madre era dotata di «una grande agilità nella manipolazione delle formule e di un meraviglioso dono per la ricerca di soluzioni molto eleganti». EDSEGER WYBE DIJKSTRA, *From my Lyfe*, EWD1166, 1993, p. 0. Il sottotitolo del manoscritto recita: "Scritto perché la gente mi chiede queste informazioni".

no, era il 1947-1948, pensava di andare all'università a studiare diritto nella speranza di rappresentare il suo Paese alle Nazioni Unite: furono gli insegnanti a consigliare gli studi scientifici. Aveva, in effetti, i voti migliori in matematica, fisica e chimica. Dijkstra ricorda che, a sedici anni, aveva acquistato i libri per l'anno successivo e si era particolarmente intimorito per il nuovo libro sulla trigonometria, pieno di seni, coseni e lettere greche. Chiese a sua madre se la trigonometria sarebbe stata difficile. La risposta avrebbe influenzato tutto il suo successivo lavoro: "Oh, no! Impara le formule e ricordati sempre che sei sulla strada sbagliata se hai bisogno di più di cinque righe"³.

3. IL CENTRO MATEMATICO

Finito il liceo, andò a studiare matematica e fisica teorica all'Università di Leiden. Il padre, che aveva letto su una rivista l'annuncio di un corso di tre settimane a Cambridge sulla programmazione per computer (si trattava dell'EDSAC⁴), gli propose di partecipare. Dijkstra pensò che fosse una buona idea nella convinzione che l'uso del computer sarebbe stato utile ad un fisico teorico. Il corso si tenne nel settembre del 1951⁵.

Considerate le nuove conoscenze acquisite, Adriaan van Wijngaarden, allora direttore del Dipartimento di calcolo al Centro Matematico di Amsterdam, gli offrì un posto di lavoro che fu accettato nel marzo del 1952⁶. Al Cen-

tro Matematico, Dijkstra avrebbe lavorato, inizialmente due giorni la settimana⁷, insieme a Bram J. Loopstra e Carel S. Scholten, alla realizzazione di computer: l'ARRA⁸, l'ARRA II, il FERTA⁹ e in particolare l'ARMAC¹⁰. Dijkstra ha scritto:

*"Nessuno di noi era stato formato come matematico. Van Wijngaarden era laureato in ingegneria meccanica, ma con una lunga esperienza nel lavoro numerico; Loopstra e Scholten studiavano fisica sperimentale presso l'Università di Amsterdam e io ancora studio fisica teorica a Leiden. Il nostro comune background era che venivamo da due delle più famose scuole secondarie del paese, van Wijngaarden e io dal Gymnasium Erasmianum, Loopstra e Scholten dal Vossius Gymnasium. Lo dico perché credo fermamente che questo quadro, che comprendeva una solida formazione in cinque lingue straniere, abbia avuto una grande influenza sul modo in cui abbiamo lavorato"*¹¹.

Nel 1952 il Centro Matematico stava realizzando il computer ARRA. Quest'ultimo "era però così inaffidabile da risultare praticamente inutilizzabile¹²" e ben presto si passò a un nuovo progetto: l'ARRA II. La descrizione funzionale dell'ARRA II è stata la prima relazione¹³ che Dijkstra ha scritto come dipendente del Centro Matematico. La relazione è rivolta all'utilizzatore della macchina e descrive ciò che essa fa e non come funziona,

³ EDSGER WYBE DIJKSTRA, *My hopes of computing science*, EWD709, 1979, p. 1.

⁴ Negli anni quaranta del Novecento, l'Università di Cambridge fu molto attiva nel settore dei computer; in particolare, per il progetto di un calcolatore a programma memorizzato detto EDSAC (Electronic Delay Automatic Calculator) basato sulle idee di Von Neumann.

⁵ In un'intervista, Dijkstra ha detto: "Tutto è iniziato nel 1951, quando mio padre mi ha consentito di frequentare un corso di programmazione a Cambridge, Inghilterra". *An Interview with Edsger W. Dijkstra*, Minneapolis, Charles Babbage Institute, 2001, p. 3.

⁶ Quando è entrato al Centro Matematico, Dijkstra aveva ventuno anni e il suo stipendio era circa l'equivalente di 3 \$ al giorno. EDSGER WYBE DIJKSTRA, *A programmer's early memories*, EWD568, 1976, p. 6.

⁷ Il resto del tempo studiava a Leiden. EDSGER WYBE DIJKSTRA, *What led to "Notes on Structured Programming"*, EWD1308, 2001, p. 0.

⁸ Automatische Relais Rekenmachine Amsterdam.

⁹ Fokker Electronische Rekenmachine te Amsterdam. La macchina fu commissionata da un'industria d'aeromobili, la Fokker. Al lavoro partecipò anche Gerard Blaauw che più tardi divenne uno dei progettisti dell'IBM 360.

¹⁰ L'acronimo sta per Automatische Rekenmachine MAthematische Centrum.

¹¹ EDSGER WYBE DIJKSTRA, *A programmer's early memories*, cit., p. 0-1.

¹² EDSGER WYBE DIJKSTRA, *A programmer's early memories*, cit., p. 1-2. Si veda anche Corrado Bonfanti: *Mezzo secolo di futuro: l'informatica italiana compie cinquant'anni*. *Mondo Digitale*, n. 3, 2004, p. 48-68: 56.

¹³ EDSGER WYBE DIJKSTRA, *Functionele beschrijving van de arra*, MR12, 1953.



Edsger Wybe
Dijkstra

quali sono le istruzioni che mette a disposizione e non come sono state realizzate. È evidente come, da subito, Dijkstra era sensibile a quelli che più tardi sarebbero stati chiamati principi di progettazione del software. L'ARMAC è del 1956. Dijkstra scrisse il software¹⁴, Loopstra e Scholten progettarono l'hardware:

*“La prima cosa che abbiamo fatto è stata quella di scrivere un manuale di programmazione per la prossima macchina, compresa la sua **completa** descrizione funzionale. (Per qualche motivo, sono stato quello che ha scritto quel documento). Questo documento quindi è servito da contratto tra me e loro: essi sapevano quello che avrebbero dovuto costruire e io sapevo quello che avrei dovuto sviluppare. E circa un anno più tardi, saremmo stati pronti: essi avevano costruito la loro macchina e io avevo finito tutti i software di base¹⁵”.*

La prima applicazione¹⁶ per l'ARMAC fu scritta da Maria C. Debets, destinata a diventare, un anno più tardi, Mrs. Dijkstra. È in questo periodo che Dijkstra prese una decisione fondamentale.

Racconta:

“Nel 1955 ho preso la decisione di non diventare un fisico teorico ma un programmatore. Ho preso questa decisione perché avevo concluso che tra la fisica teorica e la programmazione, la programmazione rappresentava una sfida intellettuale più grande. Come vedete, in quei giorni non ho sofferto di modestia intellettuale. È stata una decisione difficile, perché ero stato istruito come uno scienziato primo della classe, e diventare un programmatore sembrava un addio dalla scienza. Quando ho spiegato ad A. van Wijngaarden, il mio capo, il mio dilemma, egli mi ha assicurato che i computer erano qui per rimanere e che nel mondo della programmazione avrei benissimo potuto essere quello chiamato a creare la scienza che era ancora carente. Ottenere la laurea in fisica a Leiden divenne una formalità da sbrigare il più rapidamente possibile¹⁷”.

Nel 1956 Dijkstra si laureò in Fisica teorica. Nello stesso anno si svolse l'inaugurazione ufficiale dell'ARMAC. Per dimostrarne la potenza di calcolo, Dijkstra progettò il famoso algoritmo dei cammini minimi, ora noto come l'algoritmo di Dijkstra. Considerati i fini dimostrativi, era necessario realizzare un programma per risolvere un problema che anche i non matematici potessero capire: ad esempio, trovare l'itinerario più breve tra due città dei Paesi Bassi¹⁸. L'algoritmo fu progettato in appena ventuno minuti, senza carta e penna, mentre gustava una tazza di caffè in compagnia della futura moglie in una soleggiata terrazza di un bar di Amsterdam¹⁹.

¹⁴ Fu il primo programmatore del Centro, in un periodo in cui la professione di programmatore non era ancora molto conosciuta. Dijkstra racconta che sull'atto di matrimonio non gli fu consentito di inserire alla voce Professione il termine Programmatore. EDSGER WYBE DIJKSTRA, *The humble programmer*, “Communications of the ACM”, Vol. XV, n. 10, 1972, p. 859-866: 860.

¹⁵ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 2.

¹⁶ Si trattava del calcolo del movimento trasversale delle carrozze ferroviarie.

¹⁷ EDSGER WYBE DIJKSTRA, *What led to “Notes on Structured Programming”*, cit., p. 1.

¹⁸ Per la dimostrazione, Dijkstra selezionò 64 città e usò una codifica a 6 bit per identificare le città.

¹⁹ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 2.

Poco tempo dopo, mentre lavorava all'X1²⁰, progettò l'algoritmo per l'albero ricoprente minimo. Il problema da risolvere era quello di collegare una serie di punti sul pannello della macchina utilizzando lo stesso filo di rame in modo che tutti i punti avessero uguale tensione, e riducendo al minimo la quantità di rame usata. Così come per l'ARMAC, anche per l'X1 Dijkstra scrisse il software di base, progettando inoltre un sistema di gestione degli interrupt e fu questo l'argomento della sua tesi di dottorato²¹.

I due algoritmi furono pubblicati, in tre pagine, solo nel 1959²²:

“Non ebbi fretta di pubblicare questi due algoritmi: in quel momento, gli algoritmi non avevano ancora acquistato rispettabilità matematica e non vi erano riviste adatte. Alla fine sono stati offerti nel 1959 alla “Numerische Mathematik”, nel tentativo di aiutare questa nuova rivista ad affermarsi²³”.

Dopo Fortran, ALGOL 60 è stato uno dei primi linguaggi di programmazione ad alto livello. Dijkstra era strettamente coinvolto nello sviluppo, realizzazione e divulgazione dell'ALGOL 60. Tra il 1958 e il 1959 prese parte ad una serie di incontri che portarono alla definizione del linguaggio²⁴. Nell'estate del 1959, mentre era in vacanza con la famiglia, aveva scoperto come implementare *subroutine* ricorsive²⁵,

entro la fine dell'anno aveva definito come usare uno *stack*²⁶ per la valutazione delle espressioni e come tradurle dalla solita notazione infissa a quella polacca inversa²⁷. Tutti lavori preparatori che lo portarono nel 1960 a realizzare, in collaborazione con Jaap Zonneveld, il primo compilatore ALGOL 60²⁸:

“Abbiamo scritto il compilatore in doppia copia, pienamente consapevoli che stavamo facendo la storia: ogni sera ciascuno di noi portava la propria copia del manoscritto a casa in modo tale che, in caso d'incendio presso il nostro Centro Matematico, il compilatore non sarebbe andato perso. (Non c'è stato fuoco)²⁹”.

Il primo caso di test fu:

```
begin
  real a;
  a:=7
end.
```

Ricordando quel periodo, Dijkstra ha scritto:

“La nostra attuazione dell'ALGOL 60 segna la fine del periodo di crescita che intendevo coprire e anche l'inizio di uno nuovo. Finalmente iniziai a considerarmi un programmatore professionista [...]. Essa segnò l'inizio del periodo durante il quale l'attività di programmazione comincia ad evolvere da arte in una disciplina scientifica³⁰”.

²⁰ L'X1 fu progettato dal Centro Matematico e realizzato dall'Electrologica.

²¹ EDSGER WYBE DIJKSTRA, *Communication with an Automatic Computer*, PhD thesis, Amsterdam, 1959. Il supervisore della tesi di dottorato fu A. van Wijngaarden. Qui sono discusse le questioni concernenti la lentezza dei dispositivi di I/O rispetto al processore, è sviluppato il concetto di *interrupt* e sono descritte alcune tecniche di *buffering*.

²² EDSGER WYBE DIJKSTRA, *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. I, 1959, p. 269-271.

²³ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 2.

²⁴ Si veda KRZYSZTOF R. APT, *Edsger Wybe Dijkstra (1930-2002): A Portrait of a Genius*. Formal Aspects of Computing, Vol. XIV, 2, 2002, p. 92-98: 93.

²⁵ EDSGER WYBE DIJKSTRA, *Twenty-eight years*, EWD1000, 1987, p. 0.

²⁶ «Al quel tempo, scrivevo in olandese e avevo usato il sostantivo “stapel” e il verbo “stapelen”, che ho poi tradotto in “a stack” e “to stack”. (Evidentemente, la terminologia fu fortunata: nel giro di pochi anni dopo che la ebbi usata nell'articolo “Recursive Programming”, si è affermata nel vocabolario della comunità informatica)». *Twenty-eight years*, cit., p. 1. I termini *stack*, *vector* e *mutual exclusion*, in un contesto informatico, sono attribuiti dall'Oxford English Dictionary a Dijkstra.

²⁷ EDSGER WYBE DIJKSTRA, *Recursive Programming*. Numerische Mathematik, Vol. II, 1960, p. 312-318. Questo lavoro è considerato uno dei primi importanti contributi alla scrittura dei compilatori.

²⁸ Il lavoro durò otto mesi. Dijkstra e Zonneveld concordarono di non radersi fintanto che il progetto non fosse stato completato. Ultimato il compilatore Zonneveld si rasò, Dijkstra tenne la barba fino alla morte. Si veda FRANS E. J. KRUSEMAN ARETZ, *The Dijkstra-Zonneveld ALGOL60 Compiler for the Electrologica X1*, Centrum voor Wiskunde en Informatica, Amsterdam, 2003, p. 6.

²⁹ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 3.

³⁰ EDSGER WYBE DIJKSTRA, *A programmer's early memories*, cit., p. 11.

L'intera esperienza mi ha reso sensibile a ciò che più tardi sarebbe stata chiamata modularità o divide and rule o astrazione, per la cura con cui le interfacce furono scelte e per il potenziale campo d'applicazione della sfida della programmazione in generale. Essa ha fortemente contribuito alla mia successiva opinione che creare fiducia nella correttezza della progettazione era il più importante ma più difficile aspetto del compito del programmatore³¹.

4. L'UNIVERSITÀ TECNOLOGICA DI EINDHOVEN

Nel 1962, Dijkstra fu nominato professore nel Dipartimento di Matematica dell'Università Tecnologica di Eindhoven.

“Più tardi seppi che ero stato la terza scelta del dipartimento, dopo che due analisti avevano rifiutato l'invito; la decisione di invitare me non deve essere stata facile, da un lato perché non avevo veramente studiato matematica, e, dall'altro, a causa dei miei sandali, della mia barba e della mia “arroganza” (da vedere che cosa può essere). E' anche stato ricordato che dovevo essere molto grato al Dipartimento perché aveva preso un gran rischio a nominare me. Per due anni ho tenuto i corsi di Analisi Numerica perché non avevano nessun altro per farlo, e allo stesso tempo ho iniziato la costruzione di un piccolo gruppo di scienziati del computer³²”.

È qui, infatti, che insieme con alcuni studenti³³ progettò il THE³⁴, il primo sistema operativo con una struttura gerarchica a strati. Il sistema fu progettato per l'X8 dell'Electrologica.

Sono questi gli anni che videro Dijkstra approfondire le problematiche della programmazione concorrente³⁵. Il concetto di semaforo³⁶ e la descrizione delle operazioni P e V³⁷ sono apparsi in un manoscritto del 1962³⁸. L'intuizione del concetto nasce dall'esempio della ferrovia a binario unico: un treno prima di entrare nel binario deve attendere che lo stato del semaforo sia libero, quando entra deve modificare lo stato del semaforo per impedire che altri treni possano entrare, quando esce deve modificare ancora lo stato del semaforo per consentire agli altri treni di usare il binario. L'algoritmo del banchiere è apparso in un manoscritto probabilmente dello stesso periodo³⁹. Nel 1965, in una sola pagina⁴⁰, Dijkstra introdusse il problema della mutua esclusione per n processi e ne propose una soluzione. È stato probabilmente il primo algoritmo concorrente pubblicato. La nozione, ora standard, di sezione critica⁴¹ è stata introdotta in questo documento. Lo scritto *Cooperating sequential processes* fu realizzato nel 1965⁴² e definitivamente pubblicato nel 1968 in un saggio⁴³ di 70 pagine con cui nacque il campo della programmazione concorrente. Poi, in un documento del 1971⁴⁴ illustrò il problema del *deadlock* per mezzo dell'esempio dei “cinque filosofi a cena”. Cinque filosofi, seduti intorno ad un tavolo

³¹ EDSGER WYBE DIJKSTRA, *What led to “Notes on Structured Programming”*, cit., p. 6.

³² EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 4.

³³ Cor Ligtmans, Piet Voorhoeve, Nico Habermann e Frits Hendriks.

³⁴ Acronimo di Technische Hogeschool Eindhoven.

³⁵ Nello stesso periodo furono introdotte le “reti di Petri”, un importante contributo allo sviluppo della teoria dei processi concorrenti. CARL ADAM PETRI, *Kommunikation mit Automaten*, PhD thesis, Darmstadt, 1962.

³⁶ Il semaforo è una variabile intera non negativa condivisa tra più processi concorrenti.

³⁷ *Verhogen* sta per incrementare, *Prolagen* è un neologismo proveniente da *probeer te verlagen* e sta per testare e decrementare.

³⁸ EDSGER WYBE DIJKSTRA, *Multiprogramming en X8*, EWD51, 1962.

³⁹ EDSGER WYBE DIJKSTRA, *Een algoritme ter voorkoming van de dodelijke omarming*, EWD108, s.d.. *Un algoritmo per impedire l'abbraccio mortale*. L'abbraccio mortale (*deadlock*) è la situazione di stallo.

⁴⁰ EDSGER WYBE DIJKSTRA, *Solution of a problem in concurrent programming control*. Communications of the ACM, Vol. VIII, n. 9, 1965, p. 569.

⁴¹ La sezione critica è quella porzione di codice in cui un processo accede a variabili condivise con altri processi.

⁴² EDSGER WYBE DIJKSTRA, *Twenty-eight years*, cit., p. 2. Sono gli appunti delle lezioni tenute all'università di Eindhoven.

⁴³ EDSGER WYBE DIJKSTRA, *Cooperating sequential processes*, in *Programming Languages: NATO Advanced Study Institute*, a cura di F. Genuys, London, Academic Press, 1968, p. 43-112.

⁴⁴ EDSGER WYBE DIJKSTRA, *Hierarchical ordering of sequential processes*, EWD310, 1971.

lo per mangiare gli spaghetti, condividono solo cinque forchette. Gli spaghetti sono scivolosi e quindi ciascun filosofo si avvale di due forchette per mangiare. Purtroppo se tutti e cinque afferrano simultaneamente una forchetta, nessuno potrà mangiare. Questo problema è poi diventato un classico di riferimento per spiegare nuove primitive di sincronizzazione. Nel 1968, Dijkstra pubblicò uno dei suoi più noti articoli, *Go To statement considered harmful*⁴⁵, nel quale sostenne che la facilità di lettura e di comprensione di un programma erano inversamente proporzionali al numero di go to contenuti e che la dichiarazione go to era una delle principali fonti d'errori e pertanto andava eliminata e sostituita con strutture più evolute.

Verso la fine degli anni sessanta, Dijkstra attraversò probabilmente il momento più difficile della sua vita. Il successo del THE e la crescente autonomia della scienza informatica non erano ben visti all'interno del Dipartimento di Matematica dell'Università di Eindhoven. Il piccolo gruppo di persone che partecipò al THE fu sciolto. Dijkstra soffrì un periodo di profonda depressione. Fu così che nel 1969, "per ragioni terapeutiche"⁴⁶, iniziò a scrivere *Notes on Structured Programming*⁴⁷. Ne inviò solo una ventina di copie ad amici all'estero e ciononostante il documento ebbe una diffusione pressoché istantanea⁴⁸. La programmazione strutturata è una metodologia di scrittura dei programmi basata sulla combinazione e nidificazione di un piccolo numero di strutture di controllo⁴⁹. Dijkstra raccomandava, inoltre, alcuni principi di progettazione, che sono poi divenuti generalmente accettati nella comunità informatica:

□ i grandi sistemi devono essere costruiti partendo da molti piccoli componenti;

□ ogni componente deve essere definito solo dalla sua interfaccia e non dalla sua implementazione;

□ i componenti più piccoli possono essere progettati a seguito di un simile processo di decomposizione, determinando in tal modo un processo di tipo *top-down*;

□ la progettazione dovrebbe iniziare dalla "separazione degli interessi", cioè separando i diversi aspetti del problema e concentrando l'attenzione su ciascuno di loro separatamente;

□ la logica matematica è, e deve essere, la base per la progettazione del software.

Nel 1968 partecipò alla Conferenza sull'Ingegneria del software sponsorizzata dalla NATO Science Committee. Dijkstra fu uno dei protagonisti anche della successiva conferenza NATO che si tenne a Roma nell'ottobre del 1969 e di cui fu presidente il professor Paolo Ercoli.

Nel 1972 vinse il premio Turing dell'ACM, uno dei più importanti premi per l'Informatica. Il suo discorso d'accettazione, intitolato *The Humble Programmer*⁵⁰, affronta l'evoluzione della programmazione come disciplina e riporta una serie di prescrizioni per la sua costante crescita. Dijkstra ricorda:

"Un tardo pomeriggio nel 1972, il telefono del mio ufficio presso l'Università squillò; ho alzato il ricevitore, era Franz L. Alt dell'ACM, che chiamava dagli Stati Uniti per riferirmi che avevo vinto l'ACM Turing Award. Mi prese di sorpresa, non mi consideravo un potenziale beneficiario del prestigioso premio, e non potevo quasi crederci, ma Alt mi convinse che era proprio vero, quindi abbiamo scollegato⁵¹".

⁴⁵ EDSGER WYBE DIJKSTRA, *Go To statement considered harmful*. Communications of the ACM, Vol. XI, 3, 1968, p. 147-148. Il titolo originale del documento sottoposto era "A case against the go to statement". Al fine di accelerarne la pubblicazione, il curatore lo aveva trasformato in una lettera al Curatore modificandone anche il titolo. Il curatore era Niklaus Wirth.

⁴⁶ EDSGER WYBE DIJKSTRA, *Twenty-eight years*, cit., p. 2.

⁴⁷ EDSGER WYBE DIJKSTRA, *Notes on Structured Programming*, EWD249, 1969.

⁴⁸ «Ho inviato solo circa 20 copie ad amici all'estero, ma in quel momento la fotocopiatrice era diventata onnipresente, e (senza che lo sapessi), il testo ebbe una diffusione istantanea: più tardi ho incontrato persone in luoghi lontani che custodivano una copia di quinta o sesta generazione!». EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 5.

⁴⁹ Il fondamento teorico della programmazione strutturata è il celebre teorema di Jacopini-Böhm: qualsiasi programma scritto usando il go to poteva essere riscritto senza, a patto di avere a disposizione altri tre tipi di strutture di controllo: sequenza, iterazione e selezione. CORRADO BÖHM, GIUSEPPE JACOPINI, *Flow diagrams, turing machines and languages with only two formation rules*. Communications of the ACM, Vol. IX, 5, 1966, p. 366-371.

⁵⁰ EDSGER WYBE DIJKSTRA, *The humble programmer*. Communications of the ACM, Vol. XV, n. 10, 1972, p. 859-866.

⁵¹ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 5.

La citazione, letta dal presidente del premio, recita:

“Il dono prezioso che questo premio Turing riconosce è lo stile di Dijkstra: il suo approccio alla programmazione come una sfida alta, intellettuale; la sua eloquente insistenza e dimostrazione pratica che i programmi dovrebbero essere composti correttamente, non solo corretti in debug; e la sua illuminante percezione dei problemi alla base della progettazione dei programmi⁵²”.

5. LA BURROUGHS CORPORATION

Il 1° agosto del 1973 Dijkstra entrò, come collaboratore alla ricerca, alla Burroughs Corporation⁵³. Il suo compito consisteva nel visitare i centri di ricerca della compagnia un paio di volte l'anno e nel fare, a sua volta, ricerca, cosa che fece nello studio al secondo piano della sua casa di Nuenen, Paesi Bassi. Viaggiò molto e tenne conferenze in tutto il mondo. Fondò quello che fu chiamato l'ETAC, il **Club del Martedì Pomeriggio** ad Eindhoven: un gruppo di ricercatori⁵⁴ che incontrava tutti i martedì pomeriggio all'università per lavorare su un problema scientifico o per discutere su un documento pubblicato di recente. In questo periodo scrisse un'enorme quantità di articoli di ricerca⁵⁵, la maggior parte relazioni tecniche destinate alla circolazione privata all'interno di un gruppo selezionato. Parte di questi articoli furono pubblicati nel 1982 in una raccolta dal titolo *Selected writings on computing: a personal perspective*:

“A partire dall'estate del 1973, quando diventai Fellow alla Burroughs Research, la vita fu molto diversa da quello che era stata prima. La routine quotidiana cambiò: invece di andare al-

l'Università ogni giorno, dove ero solito trascorrere la maggior parte del mio tempo in compagnia di altri, ora andavo lì solo un giorno la settimana ed ero la maggior parte del tempo - così è, quando non si viaggia! - da solo nel mio studio. Nella mia solitudine, la corrispondenza e la parola scritta, in generale, divennero sempre più importanti. La circostanza che il mio datore di lavoro e io avessimo l'Oceano Atlantico tra noi è stato un ulteriore incentivo a tenere una traccia completa di quello che stavo facendo. La parte pubblica di tale produzione trovò il suo posto in quella che divenne nota come “la serie EWD”, che può essere vista come una forma di corrispondenza scientifica, possibile con l'avvento della fotocopiatrice. (La stessa fotocopiatrice rende difficile stimare la sua effettiva distribuzione: io feci circa due dozzine di copie dei miei testi, ma i loro destinatari accettarono di fungere come successivi nodi dell'albero di distribuzione)⁵⁶”.

Nel 1976 pubblicò la sua *A Discipline of Programming*⁵⁷. In quest'opera, un linguaggio di programmazione è visto come un veicolo per la descrizione formale e astratta di programmi indipendenti dalla macchina su cui questi devono essere eseguiti. Non sempre i programmi di una certa dimensione raggiungono una ragionevole robustezza attraverso il testing e, a volte, questi risultano poco comprensibili anche per i loro progettisti. Non si dovrebbe, quindi, prima scrivere il programma e poi sperare che sia corretto, o, solo successivamente, dimostrarne la correttezza. Dijkstra ritiene che questo metodo di costruzione di programmi sia fondamentalmente errato. Il programma e la prova di correttezza, invece, andrebbero sviluppati in parallelo: in questo modo, il programma risulterebbe corretto per costruzione. Da questo punto di vista, il fatto che un programma possa essere effettivamente ese-

⁵² EDSGER WYBE DIJKSTRA, *The humble programmer*, cit., p. 859.

⁵³ La Burroughs era una delle più importanti aziende di computer di quegli anni.

⁵⁴ Martin Rem, Maarten Boasson, Jan Tijmen Udding, Wim Feijen, Rob Hoogerwoord, Jan van de Sneyscheut, Netty van Gasteren, Anne Kaldewaij, Alain Martin, Mohamed Gouda.

⁵⁵ Alcuni lavori di questo periodo sono: *Self-stabilizing systems in spite of distributed control*, EWD391, 1973; *Guarded commands, nondeterminacy, and the formal derivation of programs*, EWD418, 1974; *On-the-fly garbage collection: an exercise in cooperation*, EWD520, 1975; *Smoothsort, an alternative for sorting in situ*, EWD796, 1981.

⁵⁶ EDSGER WYBE DIJKSTRA, *Selected writings on computing: a personal perspective*, New York-Heidelberg-Berlin, Springer-Verlag, 1982, p. V.

⁵⁷ EDSGER WYBE DIJKSTRA, *A Discipline of Programming*, London, Prentice-Hall, 1976.

guito da un computer è del tutto incidentale. Dijkstra lavorò tutta la vita alla professionalizzazione accademica della programmazione e all'autonomia dell'informatica come scienza. Autonomia non solo dalle altre scienze ma anche dal computer inteso come strumento: memorabile è la sua frase «Computer Science is no more about computers than astronomy is about telescopes».

Una disciplina accademica ha però bisogno di un oggetto e di un metodo. Insieme con un collega, Wim Feijen, scrisse *A Method of Programming*⁵⁸. In questo libro la programmazione è presentata come un ramo formale della matematica.

6. L'UNIVERSITÀ DI AUSTIN

Il periodo alla Burroughs fu intenso; tuttavia l'esigenza di un ritorno all'università era sentita. Fu così che quando l'Università del Texas di Austin gli offrì la cattedra di informatica, questa fu accettata. Era l'estate del 1984. Al riguardo, Dijkstra racconta:

L'offerta non fu una sorpresa, né la mia accettazione, conoscevo Austin, conoscevo UT e loro conoscevo me. Attraverso tutti i miei anni alla Burroughs avevo preso l'abitudine di visitare, ovunque andassi, l'università locale e sin dalla fine degli anni '70 la Burroughs aveva avuto un Centro di Ricerca ad Austin, una visita che fu inclusa nella maggior parte dei miei viaggi negli Stati Uniti⁵⁹.

Ad Austin, Dijkstra profuse gran parte delle sue energie nell'insegnamento. Non ha mai seguito un libro di testo, non ha quasi mai utilizzato proiettori, era solito invece scrivere su una lavagna.

Molti dei documenti scritti in questo periodo trattano argomenti già affrontati in passato, ne ripercorrono la storia o aggiungono nuovi contenuti: si riprendono le problematiche

concernenti la metodologia della programmazione⁶⁰ e gli algoritmi sui grafi⁶¹, si descrive l'esperienza che lo portò ad occuparsi dei Sistemi operativi⁶², quella che condusse alla Programmazione Strutturata⁶³.

In una nota del 1988, *A new science, from birth to maturity*, Dijkstra tratteggiò il percorso dell'informatica come scienza, dalla nascita alla maturità, dividendo schematicamente il periodo che va dagli anni cinquanta agli ottanta in quattro decenni. Ad ognuno di questi assegnò un'etichetta.

“Se poniamo la nascita dell'Informatica con l'avvento del computer a programma-controllato, l'argomento è di circa quattro decenni d'età. Questi sono stati decenni entusiastanti e sono grato per la fortuna di essere stato coinvolto per la maggior parte di tale periodo, grato perché questa partecipazione mi ha permesso di osservare tutto il processo di crescita da vicino [...] per amore di semplicità, ho diviso i quaranta anni in quattro decenni, ciascuno dei quali con una caratterizzazione grezza: potremmo caratterizzare gli anni cinquanta come il decennio dell'hardware, i sessanta come il decennio della sintassi, i settanta come il decennio della semantica, e gli ottanta -- a titolo di prova, essendo ancora giovane -- come il decennio della sintesi⁶⁴”.

Rifondò con un gruppo di studenti laureati, sul modello di quello di Eindhoven, il Club del Martedì Pomeriggio (ATAC)⁶⁵.

In occasione del suo sessantesimo compleanno nel 1990, il Dipartimento di Scienze organizzò un seminario di due giorni in suo onore, con oratori provenienti sia dagli Stati Uniti sia dall'Europa. Un gruppo di colleghi curò, inoltre, la pubblicazione di una raccolta di saggi dal titolo *Beauty is our business: a birthday*

⁵⁸ E. W. DIJKSTRA, W. H. J. FEIJEN, *A Method of Programming*, Boston, Addison Wesley, 1988. Il lavoro fu scritto in olandese nel 1984 col titolo *Een methode van programmeren* e solo più tardi tradotto in inglese.

⁵⁹ EDSGER WYBE DIJKSTRA, *From my Lyfe*, cit., p. 7.

⁶⁰ EDSGER WYBE DIJKSTRA, *Introducing a course on program design and presentation*, EWD1157, 1993.

⁶¹ EDSGER WYBE DIJKSTRA, *On Dijkstra's Lemma and Kruskal's Algorithm*, EWD1273, 1998.

⁶² EDSGER WYBE DIJKSTRA, *My recollections of operating system design*, EWD1303, 2001.

⁶³ EDSGER WYBE DIJKSTRA, *What led to "Notes on Structured Programming"*, EWD1308, 2001.

⁶⁴ EDSGER WYBE DIJKSTRA, *A new science, from birth to maturity*, EWD1024, 1988, p. 0.

⁶⁵ EDSGER WYBE DIJKSTRA, *The ATAC (= Austin Tuesday Afternoon Club)*, EWD927, 1985.

A short proof of one of Fermat's theorems.

Consider the n^p distinct "words" of p "characters" that can be formed from an "alphabet" of n distinct characters. Draw from each word an arrow to the word that is obtained by rotating its characters over one place to the left, i.e. from n_0, n_1, \dots, n_{p-1} an arrow is drawn to n_1, \dots, n_{p-1}, n_0 . Because p successive rotations transform a word into itself, the arrows form cycles of lengths that are divisors of p .

Hence, if p is prime, 1 and p are the only possible cycle lengths. Because a cycle of length 1 corresponds to a word all characters of which are equal, exactly n distinct words occur in a cycle of length 1. Hence the remaining $n^p - n$ words occur in cycles of length p , i.e. for any n and any prime p , $n^p - n$ is a multiple of p .

Plataanstraat 5
5671 AL NUENEN
The Netherlands

27 May 1980
prof. dr. Edsger W. Dijkstra
Burroughs Research Fellow.

FIGURA 1

Un manoscritto di
E.W. Dijkstra (Fonte:
www.cs.utexas.edu/~users/EWD/)

salute to Edsger W. Dijkstra⁶⁶. Il titolo, che a prima vista può sembrare un po' curioso, in realtà è la citazione di un articolo di Dijkstra:

"Inoltre, quando bene applicata, l'induzione matematica può portare ad argomentazioni molto compatte ed efficaci - in breve: belle! - e quando riconosciamo la battaglia contro il caos, il disordine, l'indomita complessità come una delle principali vocazioni dell'informatica, dobbiamo ammettere che Beauty is our Business⁶⁷".

Il seminario organizzato per il settantesimo compleanno ha, poi, un titolo particolarmente significativo: *In pursuit of simplicity*. Semplicità, concisione e originalità di approccio sono le caratteristiche distintive del metodo di lavoro di Dijkstra. In *A short proof of one of Fermat's theorems*⁶⁸, nel dimostrare il piccolo teorema di Fermat – quindi un problema squisitamente matematico – egli segue un metodo squisitamente "informatico" (Figura 1).

⁶⁶ Beauty is our business: a birthday salute to Edsger W. Dijkstra, a cura di W. H. J. FEIJEN, A. J. M. VAN GASTEREN, D. GRIES, J. MISRA, New York-Heidelberg-Berlin, Springer-Verlag, 1990 (Texts and Monographs in Computer Science).

⁶⁷ EDSGER WYBE DIJKSTRA, *Some beautiful arguments using mathematical induction*, EWD697, 1978, p. 1.

⁶⁸ EDSGER WYBE DIJKSTRA, *A short proof of one of Fermat's theorems*, EWD740, 1980. L'articolo apre anche il fascicolo degli Atti di un convegno A.I.C.A. (*Formal methods for software development*, Milano, marzo 1992) in cui Dijkstra fu l'ospite di spicco.

Dijkstra insegnò fino al 1999, quando andò in pensione. Morì a Nuenen, Paesi Bassi, il 6 agosto 2002.

7. I MANOSCRITTI

Dijkstra non ha mai scritto i suoi articoli utilizzando un computer. Ha sempre preferito fare affidamento sulla macchina per scrivere⁶⁹ e sulla sua penna stilografica Montblanc. Questi articoli, in gran parte mai pubblicati, sono stati poi distribuiti in maniera singolare: egli ne inviava delle copie ad alcuni amici e collaboratori che poi fungevano da successivi punti di distribuzione. Questi brevi articoli coprono un periodo di quarant'anni, sono numerati consecutivamente e sono in gran parte note tecniche, relazioni di viaggio, penetranti osservazioni, commenti pungenti, saggi. Essi raramente sono più lunghi di quindici pagine e le pagine sono numerate partendo da zero. L'ultimo, il n. 1318, è del 14 aprile 2002. Il primo dovrebbe essere del 1959. Nel mondo dell'informatica sono noti come le relazioni EWD, o semplicemente le EWDs. Così Dijkstra ricorda com'è iniziata:

“Ventotto anni fa ho scritto EWD0. Fu durante una sorta d'interregno scientifico: la mia tesi e il software di base per l'X1 erano stati completati e non ho potuto iniziare il compilatore ALGOL in quanto ALGOL 60 non era stato ancora definito. Ho fatto tutti i tipi di cose non collegate. È stato, per esempio, l'anno che ho pubblicato - finalmente! - i miei algoritmi per il cammino minimo e l'albero ricoprente minimo. È stato anche un tempo d'esplorazioni. In un determinato momento stavo lavorando a quattro o cinque diversi manoscritti. Naturalmente, le pagine d'ogni manoscritto erano state numerate; dopo che ebbi mischiato le pagine tra i manoscritti per la seconda volta, decisi che sarebbe stato meglio mettere fine alla confusione, e numerai i manoscritti: EWD0, EWD1, EWD3, EWD4. E questo è il modo in cui è iniziata⁷⁰”.

La sua scrittura a mano era così perfetta e distinta che alla fine del 1980 Luca Cardelli, allora della Digital Equipment Corporation, progettò un font “Dijkstra” per computer Macintosh. Poco dopo, Dijkstra ricevette una lettera in questo formato e pensò che fosse stata scritta a mano, ma poi ebbe notizia della creazione di questo font⁷¹.

I manoscritti erano un informale ma efficace veicolo di diffusione delle idee: «La serie EWD sembra un po' inusuale. Se è sufficientemente insolita per rappresentare un nuovo stile di fare scienza, lo sviluppo di questo stile può essere uno dei miei principali contributi⁷²».

8. CONCLUSIONI

In questo lavoro mi sono proposto di evidenziare come Edsger Wybe Dijkstra possa essere considerato uno dei protagonisti dell'informatica moderna. Il suo immenso coraggio e audacia intellettuale, e le profonde, nonché semplici ed eleganti, idee hanno cambiato il corso dell'informatica. Le sue opinioni in materia di scienza in generale e sulla ricerca, in particolare, sono state di notevole profondità e originalità. «Egli è stato un uomo con la luce nelle tenebre. Ha illuminato in pratica ogni problema che ha discusso». In breve, è stato un genio. In informatica siamo tutti i figli di Dijkstra⁷³.

Dijkstra ha attraversato, davvero, tutte le fasi dell'informatica moderna, dalla nascita alla maturità: il suo contributo alla professionalizzazione della programmazione e all'autonomia dell'informatica come scienza è stato decisivo. Nessun aspetto della disciplina è stato trascurato: dalla progettazione degli algoritmi ai sistemi operativi, dai linguaggi di programmazione ai compilatori ecc.. A lui dobbiamo l'intuizione che la logica e la metodologia matematica dovessero essere la base per la ragionevole progettazione dei programmi. Al suo pionieristico contributo dobbiamo la costruzione di una

⁶⁹ La macchina era una Hermes.

⁷⁰ EDSGER WYBE DIJKSTRA, *Twenty-eight years*, cit., p. 0.

⁷¹ KRZYSZTOF R. APT, *op. cit.*, p. 94.

⁷² EDSGER WYBE DIJKSTRA, *Twenty-eight years*, cit., p. 7.

⁷³ KRZYSZTOF R. APT, *op. cit.*, p. 98.

0

solida base scientifica e teorica per la scienza informatica.

A lui dobbiamo non solo l'arricchimento del vocabolario dell'informatico (stack, semafori, programmazione strutturata ecc.), ma anche la concezione della programmazione come una sfida intellettuale all'interno di un modello chiaro, semplice e matematicamente rigoroso. A lui dobbiamo l'enfasi posta sulla qualità e correttezza dei programmi (o come preferiva chiamarla: "eleganza").

«Semplicità, bellezza ed eloquenza sono state le sue caratteristiche distintive, e la sua insistenza senza compromessi sull'eleganza nella programmazione e sulla matematica è stata una fonte d'ispirazione per migliaia»⁷⁴.

⁷⁴ R. S. BOYER, W. FEIJEN, D. GRIES, C. A. R. HOARE, J. MISRA, J. MOORE, H. RICHARDS, In Memoriam Edsger W. Dijkstra 1930-2002, «Communications of the ACM», vol. VL, 10, 2002, pp. 21-22: 22.

Bibliografia

- [1] DIJKSTRA E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. I, 1959, p. 269-271.
- [2] DIJKSTRA E.W.: Go To statement considered harmful. *Communications of the ACM*, Vol. XI, n. 3, 1968, p. 147-148.
- [3] DIJKSTRA E.W., DAHL O.J., HOARE C. A.R.: *Structured Programming*. London-New York, Academic Press, 1972.
- [4] DIJKSTRA E.W.: The humble programmer. *Communications of the ACM*, Vol. XV, 10, 1972, p. 859-866.
- [5] APT KRZYSZTOF R., Edsger Wybe Dijkstra (1930-2002): A Portrait of a Genius. *Formal Aspects of Computing*, Vol. XIV, n. 2, 2002, p. 92-98.

I manoscritti della serie EWD sono disponibili sul sito web www.cs.utexas.edu/users/EWD/ curato dal Dipartimento di Informatica dell'Università del Texas.

Nota. Poiché non risultano esistere traduzioni italiane delle opere di Dijkstra, i passi riportati nell'articolo sono stati tradotti per l'occasione.

1

LORENZO MILONE è laureato in economia e commercio, lavora professionalmente nel settore del software ed è un cultore di storia dell'informatica.

E-mail: lorenzomilone@libero.it