# Un numero primo di domande al Professore Emerito dell'Arte della Programmazione: intervista a Donald Knuth

*Gianluca Pignalberi*


## Sommario

Proponiamo qui la traduzione italiana dell'intervista a Donald Ervin Knuth, apparsa per la prima volta su Free Software Magazine n. 7 e ripubblicata su TUGboat Volume 26, Number 6.


La composizione della rivista Free Software Magazine è interamente basata su TEX. Forse qualcuno non sa ancora che il Professor Donald Knuth ha progettato TEX e l'ha fatto circa 30 anni fa. Da allora il progetto TEX ha generato molti strumenti correlati (ad es., LATEX, ConTEXt, Ω e altri).

Nel 2005 ho avuto l'opportunità e l'onore di intervistare il professor Knuth. Sono orgoglioso, come giornalista e TEXnico di FSM, di vedere l'intervista ripubblicata sulla prima rivista italiana di TEX e LATEX.

*Donald E. Knuth, Professore Emerito dell'Arte della Programmazione, professore di matematica (concreta), creatore di TEX e* METAFONT*, autore di diversi libri fantastici (come* Computers and Typesetting*,* The Art of Computer Programming*,* Matematica discreta[1]*) e articoli, assegnatario del* Turing Award*, del* Kyoto Prize *e altri importanti riconoscimenti; membro della* Royal Society *(e potrei andare avanti). C'è qualche argomento che avrebbe voluto padroneggiare e non l'ha fatto? Se sì, perché?*

Grazie per le sue parole gentili, ma in realtà provo costantemente ad imparare nuove cose nella speranza di poter poi aiutare ad insegnarle ad altri. Vorrei inoltre poter capire lingue diverse dall'inglese senza troppa difficoltà; sono spesso limitato dalla mia incompetenza linguistica, mentre voglio capire la gente di altre culture ed altre ere.

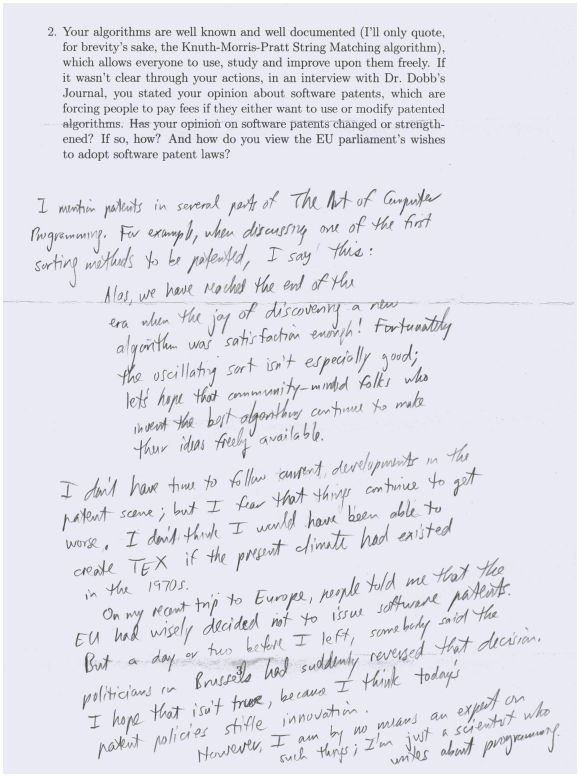
*I suoi algoritmi sono ben conosciuti e ben documentati (citerò solamente, per amor di brevità, l'algoritmo di string matching Knuth-Morris-Pratt), cosa che permette a chiunque di usarli, studiarli e migliorarli liberamente. Se non fosse chiaro dalle sue azioni, in un'intervista al Dr. Dobb's Journal ha dichiarato la sua opinione circa i brevetti software, che costringono la gente a pagare delle quote qualora vogliano usare o modificare algoritmi brevettati. La sua opinione sui brevetti software è cambiata o si è rafforzata? In che modo? E come vede i desideri del Parlamento Europeo di adottare leggi sui brevetti software?*

Menziono i brevetti in diverse parti di *The Art of Computer Programming*. Per esempio, discutendo uno dei primi metodi di ordinamento ad essere brevettato, dico questo:

> Purtroppo, abbiamo raggiunto la fine dell'era in cui la gioia di scoprire un nuovo algoritmo era una soddisfazione sufficiente! Fortunatamente l'ordinamento oscillante non è particolarmente efficace; speriamo che le persone orientate alla comunità


FIGURA 1: Il prof. Knuth mentre legge una delle riviste composte dal suo programma TEX. Foto di Jill Knuth (laureata al Flora Stone Mather College — un altro FSM)

1. È il poco significativo titolo italiano di *Concrete Mathematics*, dove Concrete è la contrazione di '**con**tinuous' e 'dis**crete**'


2. Your algorithms are well known and well documented (I'll only quote, for brevity's sake, the Knuth-Morris-Pratt String Matching algorithm), which allows everyone to use, study and improve upon them freely. If it wasn't clear through your actions, in an interview with Dr. Dobb's Journal, you stated your opinion about software patents, which are forcing people to pay fees if they either want to use or modify patented algorithms. Has your opinion on software patents changed or strengthened? If so, how? And how do you view the EU parliament's wishes to adopt software patent laws?

I mention patents in several parts of The Art of Computer Programming. For example, when discussing one of the first sorting methods to be patented, I say this:

Alas, we have reached the end of the era when the joy of discovering a new algorithm was satisfaction enough! Fortunately the oscillating sort isn't especially good; let's hope that community-minded folks who invent the best algorithms continue to make their ideas freely available.

I don't have time to follow current developments in the patent scene; but I fear that things continue to get worse. I don't think I would have been able to create TEX if the present climate had existed in the 1970s.

On my recent trip to Europe, people told me that the EU had wisely decided not to issue software patents. But a day or two before I left, somebody said the politicians in Brussels had suddenly reversed that decision. I hope that isn't true, because I think today's patent policies stifle innovation.

However, I am by no means an expert on such things; I'm just a scientist who writes about programming.

FIGURA 2: La porzione di pagina su cui Knuth ha risposto alla seconda domanda di quest'intervista. Prego notate che cita e compone tipograficamente anche quando scrive a mano: la citazione da TAOCP e la parola TEX

> che inventano i migliori algoritmi continuino a rendere le loro idee liberamente disponibili.

Non ho tempo per seguire gli attuali sviluppi nell'ambito dei brevetti; ho paura però che le cose continuino a peggiorare. Non credo che sarei stato in grado di creare TEX se il clima odierno ci fosse stato negli anni 70.

Nel mio recente viaggio in Europa, la gente mi ha raccontato che la UE aveva saggiamente deciso di non imporre i brevetti software. Ma un giorno o due prima che io ripartissi, qualcuno disse che i politici a Bruxelles avevano improvvisamente ribaltato la loro decisione. Spero che non sia vero, perché penso che le odierne politiche dei brevetti soffochino l'innovazione.

Comunque, io non sono affatto un esperto di queste cose; sono solo uno scienziato che scrive di programmazione.

*Finora lei ha scritto tre volumi di* The Art of Computer Programming*, sta lavorando al quarto, spera di finire il quinto volume entro il 2010, e ancora pianifica di scrivere i volumi 6 e 7. Esclusa la serie* Selected papers*, ci sono altri argomenti di cui lei ritiene di dover scrivere, ma non ne ha avuto il tempo?, Se sì, vuole riassumere di quale argomento parlerebbe?*

Sto facendo progressi lenti ma costanti sui volumi 4 e 5. Ho anche molte annotazioni per i volumi 6 e 7, ma questi libri trattano argomenti meno fondamentali e potrei scoprire che c'è poco bisogno di questi libri quando sarò arrivato a quel punto.

Ho paura dei 20 anni di lavoro necessari prima che io porti TAOCP ad una conclusione di successo. Dato che ho 67 anni ora, spero vivamente di stare in salute ed essere in grado di fare un buon lavoro pur invecchiando ed essendo ancor più decrepito. Fortunatamente, al momento mi sento bene come sempre.

Se avessi tempo per qualcos'altro, mi piacerebbe comporre musica. Naturalmente non so se ciò avrebbe successo; terrei la cosa per me se non fosse buona. Tuttora sento il bisogno di provare ogni tanto e i computer rendono queste cose più semplici.

*Circolano voci che lei ha iniziato il progetto TEX perché stanco di vedere i suoi manoscritti maltrattati dall'American Mathematical Society. Allo stesso tempo, afferma di aver scritto TEX dopo aver visto le prove di stampa del libro* The Art of Computer Programming*. Per piacere, racconti brevemente ai nostri lettori cosa l'ha fatta decidere ad iniziare il progetto, quali strumenti ha usato, e quante persone facevano parte del nucleo della squadra di TEX.*

No, le società matematiche non potevano essere incolpate per lo stato pietoso della tipografia nel 1975. La cosa è dovuta al fatto che le industrie della stampa erano passate a nuovi metodi e i nuovi metodi erano progettati per essere validi per le riviste, i giornali e i romanzi ma non per la scienza. Gli scienziati non avevano alcun potere economico, così a nessuno importava se i nostri libri e articoli sembravano belli o brutti.

Racconto l'intera storia nel capitolo 1 del mio libro *Digital Typography* che è ovviamente un libro che spero tutti leggano e apprezzino.

Gli strumenti che ho usato sono stati tutti autoprodotti e sono diventati famosi come *Programmazione Colta*[2]. Sono enormemente influenzato dalla programmazione colta che è sicuramente la cosa migliore mai inventata. Io continuo ad usarla per scrivere programmi quasi ogni giorno e mi aiuta ad ottenere codice efficiente, robusto e manutenibile con maggior successo di ogni altro metodo che conosco. Naturalmente, capisco che altre persone potrebbero ritenere altri approcci più di loro gusto; ma, wow, io amo gli strumenti che ho ora. Non avrei potuto affatto scrivere programmi difficili come il metasimulatore MMIX se non avessi avuto la programmazione colta; il compito sarebbe stato troppo arduo.

2. Traduzione letterale di *Literate Programming*, preferita ad altre traduzioni su una mailing list di `linux.it`

FIGURA 3: Ritratto di Donald E. Knuth di Alexandra Drofina. Gli utenti commerciali dovrebbero scrivere a Yura Revich (`revich@computerra.ru`) per il permesso

Nel nucleo della squadra di TEX avevo assistenti che leggevano il codice che scrivevo, che preparavano i driver per le stampanti e le interfacce e li portavano su altri sistemi. Ho avuto due studenti che hanno inventato gli algoritmi per la sillabazione e l'interruzione di riga. E ho avuto molte dozzine di volontari che si riunivano ogni venerdì per diverse ore per aiutarmi a prendere le decisioni. Ma ho scritto da me ogni singola linea di codice.

Il capitolo 10 del mio libro *Literate Programming* spiega perché penso che un progetto di prima generazione come questo sarebbe fallito se avessi provato a delegare il lavoro.

*Forse lei ritiene che qualcuna delle tecnologie attuali è ancora insoddisfacente. Se non fosse impegnato a scrivere i suoi capolavori, quale tecnologia proverebbe a rivoluzionare e in che modo?*

Be', certamente proverei a lavorare per la pace e la giustizia nel mondo. Tendo a pensare a me stesso come un cittadino del mondo; sono piacevolmente eccitato quando vedo il mondo diventare sempre più piccolo e la gente di diverse culture lavorare insieme rispettando le proprie differenze. Al contrario sono addolorato quando vengo a sapere dell'odio profondo o quando vedo la gente sfruttare gli altri o scacciarli preventivamente.

In che modo può accadere la rivoluzione desiderata? Chissà... ma sospetto che "Ingegneri senza frontiere" siano più vicini di chiunque altro ad una strategia funzionante mediante cui i tecnologi come me possono essere d'aiuto.

*Grazie ancora per il suo tempo prezioso.*

Grazie per avermi posto domande eccellenti!

▷ Gianluca Pignalberi
Free Software Magazine
`g.pignalberi@`
`freesoftwaremagazine.com`