L'algoritmo di Fibonacci per moltiplicare due numeri interi

La scrittura posizionale di un numero, mentre permette di scrivere (e leggere) numeri anche molto grandi, come ha spiegato Fibonacci nel primo capitolo, pone, nello stesso tempo, il problema di come eseguire le operazioni elementari dell'aritmetica. Sorprendentemente, e in modo geniale, Fibonacci, anziché iniziare con un algoritmo per sommare gli interi, inizia con il prodotto illustrando, a partire da casi semplici fino ad arrivare al caso generale, come poter moltiplicare due qualsiasi numeri interi sapendo solo quale sia il risultato che si ottiene moltiplicando due cifre, riducendo così il prodotto di due numeri con molte cifre al prodotto di due numeri di una sola cifra, in definitiva riducendo tutto alla tavola pitagorica, da tenere bene a mente e *«sulle dita»* **(I.13)**. La rappresentazione dei numeri sulle dita delle mani è una pratica molto antica [vedi scheda *Calcolare con le mani*] che Fibonacci non trascura ed anzi sollecita spesso come strumento per facilitare la memoria a ricordare i "riporti" che via via intervengono durante l'esecuzione dell'algoritmo. Il termine algoritmo è in questo caso particolarmente opportuno, non solo per il richiamo al matematico arabo al-Khwarizmi che, oltre a fondare l'algebra, ha sviluppato il calcolo indiano, noto in occidente col nome generico di algorismus [A. Allard, *Muhammad ibn Musa al Khwarizmi. Le calcul indien (Algorismus)*, Albert Blanchard, Paris, 1992], ma anche e proprio per la forma algoritmica del procedimento descritto da Fibonacci che permette di trovare le varie cifre del prodotto attraverso un processo ricorsivo facilmente riproducibile con un qualsiasi linguaggio di programmazione.

La scelta di iniziare col prodotto anziché con la somma introduce subito il lettore nel merito della rivoluzione che si stava compiendo, presentando problemi difficili se non impossibili da risolversi coi vecchi metodi, rivoluzione che dovette avere un impatto straordinario non solo su Fibonacci, che non si astiene dal confessare che il calcolo basato *«sulle nove figure degli Indiani mi piacque sopra ogni altra cosa»* **(I.3)**, ma anche sui suoi contemporanei, sui grandi mercanti pisani che con quei nuovi strumenti potevano più facilmente navigare nel mare dei loro complicati conti. E' con grande soddisfazione, mista di entusiasmo, che Fibonacci dice *«affinché gli inesperti abbiano qui un insegnamento perfetto, ho deciso di mostrare la moltiplicazione dell'ottava posizione»* **(II.40)** e spiega, a titolo di esempio, come eseguire la moltiplicazione 12.345.678 per 87.654.321, cioè come sommare 12.345.678 volte il numero 87.654.321!

Cercheremo nel seguito di illustrare l' algoritmo di Fibonacci per la moltiplicazione con i simboli e il linguaggio della matematica contemporanea, per renderlo esplicito a un lettore di oggi, ma anche per evidenziare la modernità del metodo e il suo carattere ricorsivo che permette la scrittura di un semplice algoritmo, che proponiamo in una apposita *scheda* in Python. La nostra presentazione, a differenza di quella di Fibonacci, che dal semplice arriva al generale, non è didattica ma è pensata per gli insegnanti di Matematica o Informatica in modo che, una volta compresa la logica dell'algoritmo, possano trovare una loro forma didattica, eventualmente seguendo il percorso che lo stesso Fibonacci propone nel secondo capitolo del *Liber abaci*.

Usando la scrittura decimale, un numero intero di n cifre: $\boldsymbol{a} = a_{n-1}...a_3\, a_2\, a_1\, a_0$ $(0 \leq a_i < 10)$ è dato dalla somma:

$$\boldsymbol{a} = a_0 + a_1 10 + a_2 10^2 + a_3 10^3 + ... + a_{n-1} 10^{n-1}$$

e un secondo numero di m cifre: $\boldsymbol{b} = b_{m-1}..b_3\, b_2\, b_1\, b_0$ dalla somma:

$$\boldsymbol{b} = b_0 + b_1 10 + b_2 10^2 + b_3 10^3 + ... + b_{m-1} 10^{m-1}$$

e il loro prodotto di ottiene usando più volte la proprietà associativa, commutativa e distributiva del prodotto rispetto all'addizione. La sola che Fibonacci esplicitamente cita, dando le altre per scontate è quella distributiva:

*«...quando un qualche numero è diviso in parti e ogni singola parte è moltiplicata per un qualche numero, quelle moltiplicazioni raccolte in una sola sono uguali alla moltiplicazione di tutto il numero diviso, per il numero per il quale furono moltiplicate tutte le parti dello stesso.»***(II.7)**

Si tratta dunque di calcolare, distribuendo i fattori, i numeri A_0 , A_1 , ... A_{n+m-2} che compaiono nel prodotto:

$$\boldsymbol{a}\, x\, \boldsymbol{b} = (a_0 + a_1 10 + ... + a_{n-1} 10^{n-1}) \times (b_0 + b_1 10 + ... + b_{m-1} 10^{m-1}) =$$
$$= A_0 + A_1\, 10 + A_2\, 10^2 + ... + A_{n+m-2}\, 10^{n+m-2}$$

Tali numeri A_k sono somme di prodotti di due numeri di una cifra ciascuno, sono cioè somme di un certo numero di numeri di una o due cifre. Queste somme,

nell'impostazione di Fibonacci, sono calcolate direttamente, a mente o con le dita, senza bisogno di un algoritmo speciale per poterle trovare.

Un esempio alla lavagna

Per renderci conto della situazione e per organizzare correttamente gli indici eseguiamo esplicitamente e per disteso l'algoritmo nel caso di due numeri $\boldsymbol{a}$ e $\boldsymbol{b}$ di tre e due cifre.

Fibonacci indica di scrivere in *«una tavola pulita nella quale le lettere si cancellano facilmente»* **(II.3)** i due numeri in modo che le posizioni si corrispondano: il numero maggiore sotto il minore, cosa che implica implicitamente l'assunzione della proprietà commutativa del prodotto.

Eseguiamo ora il prodotto applicando la proprietà distributiva

$$\boldsymbol{a} \times \boldsymbol{b} = (a_0 + a_1 10 + a_2 10^2) \times (b_0 + b_1 10) =$$
$$= a_0 \times b_0 + (a_0 \times b_1 + a_1 \times b_0)10 + (a_1 \times b_1 + a_2 \times b_0)10^2 + a_2 \times b_1\ 10^3 =$$
$$= A_0 + A_1 10 + A_2 10^2 + A_3 10^3$$

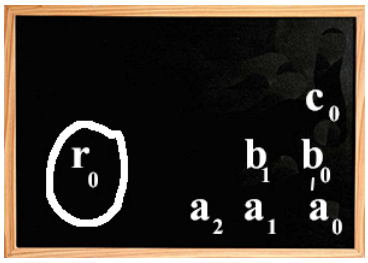
Il numero $A_0 = a_0\ b_0$ è un prodotto di due cifre, il numero di unità di $\boldsymbol{a}$ e quello di $\boldsymbol{b}$, che sarà trovato a memoria o utilizzando la tavola pitagorica. Questo prodotto, come ogni altro numero intero, sarà formato da un certo numero c_0 di unità e da un multiplo di 10:

$$a_0 \times b_0 = A_0 = c_0 + r_0 10 \quad \text{dove} \quad 0 \leq c_0 < 10 \quad \text{e} \quad 0 \leq r_0$$

$$\boldsymbol{a} \times \boldsymbol{b} = a_0 \times b_0 + (a_0 \times b_1 + a_1 \times b_0)10 + \ldots = c_0 + r_0 10 + (a_0 \times b_1 + a_1 \times b_0)10 + \ldots =$$
$$= c_0 + (r_0 + a_0 \times b_1 + a_1 \times b_0)10 + \ldots$$

E, poiché $c_0 < 10$ sarà la prima cifra decimale del prodotto.

Scriviamo c_0 in corrispondenza delle unità e teniamo in memoria l'eventuale riporto r_0 di decine o, per non sbagliare, come consiglia Fibonacci, *«si tengano in mano le decine»* **(II.2)**. Nella nostra lavagna segniamo il riporto in un cerchio che poi cancelleremo per mettere il riporto successivo.



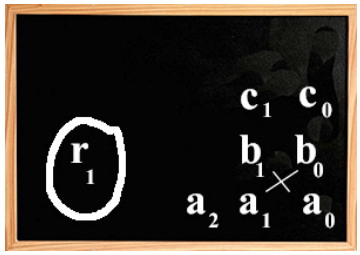
Calcoliamo ora la seconda cifra, quella delle decine, che si ottiene da A_1 che è la somma dei prodotti "in croce" con l'aggiunta del riporto r_0:

$$r_0 + a_0 \times b_1 + a_1 \times b_0 = r_0 + A_1 = c_1 + r_1 10 \quad \text{dove} \quad 0 \leq c_1 < 10 \quad \text{e} \quad 0 \leq r_1$$

$$a \times b = c_0 + (r_0 + a_0 \times b_1 + a_1 \times b_0)10 + (a_1 \times b_1 + a_2 \times b_0)10^2 + \ldots =$$
$$= c_0 + (c_1 + r_1\, 10)10 + (a_1 \times b_1 + a_2 \times b_0)10^2 + \ldots =$$
$$= c_0 + c_1\, 10 + (r_1 + a_1 \times b_1 + a_2 \times b_0)10^2 + \ldots =$$

scriviamo la seconda cifra del prodotto c_1 nel posto delle decine e teniamo in memoria il riporto r_1.

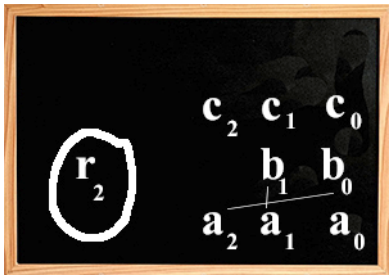
A questo punto si procede in modo analogo: calcoliamo il numero di centinaia tendo conto del riporto r_1 .



$$a_1 \times b_1 + a_2 \times b_0 + r_1 = A_2 + r_1 = c_2 + r_2 10 \quad \text{dove} \quad 0 \leq c_2 < 10 \quad \text{e} \quad 0 \leq r_2$$

$$a \times b = c_0 + c_1 10 + (c_2 + r_2 10)10^2 + a_2 \times b_1\, 10^3 =$$
$$= c_0 + c_1 10 + c_2\, 10^2 + (r_2 + a_2 \times b_1)10^3$$

Scriviamo la terza cifra del prodotto c_2 nel posto delle centinaia e teniamo in memoria il riporto r_2.

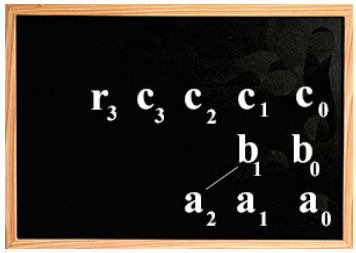


Infine

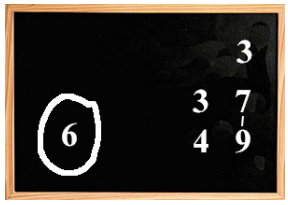
$$a_2 \times b_1 + r_2 = A_3 + r_2 = c_3 + r_3 10 \quad \text{dove} \quad 0 \leq c_3 < 10 \quad \text{e} \quad 0 \leq r_3$$

$$a \times b = c_0 + c_1 10 + c_2\, 10^2 + (c_3 + r_3 10)10^3 =$$
$$c_0 + c_1 10 + c_2\, 10^2 + c_3 10^3 + r_3 10^4$$

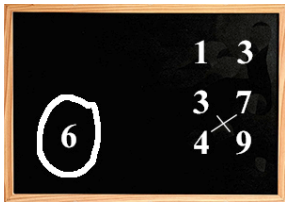
scriviamo c_3 e, se non nullo, le cifre che compongono r_3 . Ora il prodotto è completo.



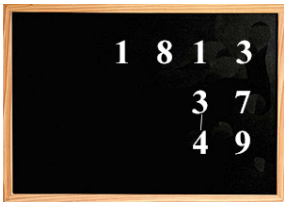
Ecco come procede il calcolo nella moltiplicazione di 37 per 49:



7x9 = 3 + 60



3x9 + 7x4 + 6 =1+ 60



3x4 + 6 = 18

Il caso generale

Consideriamo, in generale, due numeri interi a e b e il loro prodotto $a \times b$. Scrivendo i numeri col sistema decimale introdotto da Fibonacci, il prodotto dei due numeri si ottiene sviluppando, come abbiamo detto, con la proprietà distributiva, le seguenti somme:

$$a \times b = (a_0 + a_1 10 + \ldots + a_{n-1} 10^{n-1}) \times (b_0 + b_1 10 + \ldots + b_{m-1} 10^{m-1}) =$$
$$= A_0 + A_1\, 10 + A_2\, 10^2 + \ldots + A_{n+m-2}\, 10^{n+m-2} =$$
$$= c_0 + c_1 10 + c_2 10^2 + \ldots + c_p 10^{p-1}$$

I numeri A_k sono delle somme di prodotti di due numeri di una cifra e si calcolano a partire dalle singole cifre di a e b. Precisamente:

$$\mathbf{A}_k = \sum_{i+j=k} a_i \times b_j$$

in particolare $A_0 = a_0 \times b_0$, $A_1 = a_0 \times b_1 + a_1 \times b_0$, ... , $A_{n+m-2} = a_{n-1} \times b_{m-1}$.

Dato un numero intero s indichiamo con $\mathbf{u}(s)$ la sua prima cifra decimale e con $\mathbf{m}(s)$ il suo "riporto" cioè il numero ottenuto da s levando tale cifra. Così, ad esempio $\mathbf{u}(453)= 3$ e $\mathbf{m}(453)= 45$. Ogni numero s sarà quindi decomposto in due fattori: la sua prima cifra decimale e il suo riporto.

$$s = \mathbf{u}(s) + \mathbf{m}(s)10 \quad \text{con } 0 \leq \mathbf{u}(s) < 10$$

L'algoritmo che permette di calcolare le cifre c_k del prodotto $\boldsymbol{a} \times \boldsymbol{b}$ si ottiene calcolando via via i riporti r_k "da ritenere sulle dita" e questi vengono calcolati con la seguente formula ricorsiva:

$$\begin{cases} r_{-1}=0 \\ r_k = m(A_k + r_{k-1}) \end{cases} \quad (k=0,1,\ldots,n+m-2)$$

Fatto questo, le cifre decimali del prodotto sono date da $c_k = \mathbf{u}(A_k + r_{k-1})$.

Dimostrazione

Sia

$$\boldsymbol{a} \times \boldsymbol{b} = A_0 + A_1\, 10 + A_2\, 10^2 + A_3\, 10^3 + \ldots =$$

$$= [\mathbf{u}(A_0) + \mathbf{m}(A_0)10] + A_1\, 10 + A_2\, 10^2 + \ldots = \mathbf{u}(A_0) + [\mathbf{m}(A_0)+A_1]10 + A_2\, 10^2 + \ldots$$

essendo $\mathbf{u}(A_0)<10$ abbiamo $c_0 = \mathbf{u}(A_0)$ e, per la formula ricorsiva, $\mathbf{m}(A_0) = \mathbf{m}(A_0 + r_{-1}) = r_0$. Otteniamo quindi

$$\boldsymbol{a} \times \boldsymbol{b} = c_0 + (r_0+A_1)10 + A_2\, 10^2 + A_3\, 10^3 + \ldots$$

Applichiamo la stressa scomposizione al termine r_0+A_1

$$c_0 + (r_0+A_1)10 + A_2\, 10^2 + \ldots = c_0 + [\mathbf{u}(r_0+A_1) + \mathbf{m}(r_0+A_1)10]10 + A_2\, 10^2 + \ldots =$$

$$= c_0 + \mathbf{u}(r_0+A_1)10+[\mathbf{m}(r_0+A_1) + A_2\,]10^2 + A_3\,10^3 + \ \ldots$$

essendo $\mathbf{u}(r_0+A_1)<10$ abbiamo $c_1 = \mathbf{u}(r_0+A_1)$ e, dalla formula ricorsiva $\mathbf{m}(r_0+A_1) = r_1$ dunque

$$\boldsymbol{a\,x\,b} = c_0+c_1 10 +(r_1+ A_2)10^2 + A_3\,10^3 + \ \ldots$$

Applichiamo la stessa scomposizione al termine r_1+A_2

$$c_0+c_1 10 +(r_1+ A_2)10^2 + A_3\,10^3 + \ \ldots= c_0+c_1 10 + [\mathbf{u}(r_1+ A_2)+\mathbf{m}(r_1+ A_2)10]10^2 + A_3\,10^3 + \ \ldots=$$

$$= c_0+c_1 10 + \mathbf{u}(r_1+ A_2)\,10^2 +[\mathbf{m}\,(r_1+ A_2) + A_3\,]\,10^3 + \ \ldots$$

essendo $\mathbf{u}(r_1+A_2)<10$ abbiamo $c_2 = \mathbf{u}(r_1+A_2)$ e, dalla formula ricorsiva, $\mathbf{m}(r_1+A_2) = r_2$ dunque

$$\boldsymbol{a\,x\,b} = c_0+c_1 10 +c_2 10^2 + (r_2 + A_3)10^3 + \ \ldots=$$

Si capisce come il procedimento continui ricorsivamente fino alla ultima espressione

$$\boldsymbol{a\,x\,b} = c_0 + c_1 10 + c_2 10^2 + \ldots + c_{n-m-3}10^{n+m-3} + \ (r_{n+m-3} + A_{n+m-2})10^{n+m-2}$$

$$r_{n+m-3} + A_{n+m-2} = \mathbf{u}(r_{n+m-3} + A_{n+m-2})+\mathbf{m}(r_{n+m-3} + A_{n+m-2})10 = c_{n+m-2} + r_{n+m-2}\,10.$$

Sostituendo otteniamo

$$\boldsymbol{a\,x\,b} = c_0 + c_1 10 + c_2 10^2 + \ldots + c_{n-m-3}10^{n+m-3} + \ c_{n+m-2}10^{n+m-2} + r_{n+m-2}10^{n+m-1}$$

A questo punto le cifre decimali di r_{n+m-2} saranno le ultime cifre decimali del prodotto.

Un possibile miglioramento formale dell'algoritmo consiste nel definire tutti i termini A_k come somme dello stesso numero di termini. Questo è possibile ponendo $b_i = 0$ per $i > m-1$ e per $i < 0$. In questo modo risulta per ogni $k \geq 0$:

$$A_k = a_0 \times b_k + a_1 \times b_{k-1} + a_2 \times b_{k-2} + \ \ldots + a_{n-1} \times b_{k-n+1}$$

infatti:

$$A_0 = a_0 \times b_0 + a_1 \times b_{-1} + a_2 \times b_{-2} + \ \ldots + a_{n-1} \times b_{-n+1} = a_0 \times b_0,$$
$$A_1 = a_0 \times b_1 + a_1 \times b_0 + a_2 \times b_{-1} + \ \ldots + a_{n-1} \times b_{-n+2} = a_0 \times b_1 + a_1 \times b_0$$

.....

$$A_{n+m-2} = a_0 \times b_{n+m-2} + a_1 \times b_{b+m-3} + a_2 \times b_{k-2} + \ \ldots + a_{n-2} \times b_m + a_{n-1} \times b_{m-1} = \ a_{n-1} \times b_{m-1}$$

e, infine $A_k = 0$ per k >n+m-2, essendo nulli tutti i coefficienti b_k dato che k-n+1> m-1 per k > n+m-2. Questa estensione ci permette di iterare il nostro algoritmo fino a quando r_k >0. La nostra formula ricorsiva

$$\begin{cases} r_{-1}=0 \\ r_k=m\left(A_k+\mathrm{r}_{\mathrm{k-1}}\right) \end{cases}$$

ci fornisce $r_{n+m-1} = \mathbf{m}(r_{n+m-2})$ da cui $r_{n+m-2} = \mathbf{u}(r_{n+m-2})+ \mathbf{m}(r_{n+m-2})\ 10 = \mathbf{u}(r_{n+m-2})+ \ r_{n+m-1}\ 10$ e, tornando al nostro prodotto,

$$\boldsymbol{a}\ \boldsymbol{x}\ \boldsymbol{b} = c_0 + c_1 10 \ + \ldots + \ c_{n+m-2}10^{n+m-2} + r_{n+m-2}10^{n+m-1} =$$

$$= c_0 + c_1 10 \ + \ldots + \ c_{n+m-2}10^{n+m-2} + [\mathbf{u}(r_{n+m-2})+ \ r_{n+m-1}\ 10]\ 10^{n+m-1} =$$
$$= c_0 + c_1 10 \ + \ldots + \ c_{n+m-2}10^{n+m-2} + c_{n+m-1}\ 10^{n+m-1} + \ r_{n+m-1}\ 10^{n+m}$$

e l'algoritmo continua fino a quando il riporto r_k si annulla.

Si vede che da un punto di vista informatico la scrittura di questo algoritmo richiede la capacità di scrivere delle funzioni dirette o ricorsive. Si deve intanto avere una funzione *cifre_a*(i) che assegna per ogni valore dell'indice i la i-esima cifra del numero $\boldsymbol{a}$ e ugualmente, per il numero $\boldsymbol{b}$, la funzione *cifre_b*(i) estesa come detto per valori negativi di i e per valori i>m-1. Poi si deve saper definire la funzione *A*(k) che è la somma di n termini essendo n stesso una variabile e infine la funzione ricorsiva che calcola i resti. L'operazione **u**(s) è già definita in Python prendendo il resto di s modulo 10 quindi **u**(s)=s%10 e di conseguenza l'operazione **m**(s)=(s-s%10)/10.
Nella sezione informatica sarà presentato nei dettagli questo programma
[*moltiplicazione_X_2_cifre.py* , *moltiplicazione_Fibonacci.py*] e una sua variante più sintetica ed efficiente [*fibo_prodottoNxN.py*] sarà discusso nei dettagli questo programma e alcune sue possibili varianti.

Osserviamo che, quando i numeri da moltiplicare hanno molte cifre, vi è una difficoltà pratica nell'eseguire l'algoritmo di Fibonacci che consiste soprattutto nelle carenze, oggi più che mai evidenti, della memoria di lavoro. Si tratta infatti, per calcolare i numeri A_k di svolgere a mente operazioni anche piuttosto lunghe, cosa impossibile senza una buona memoria di lavoro, e di tenere a mente il resto precedente da aggiungere alla fine questa operazione. Non a caso Fibonacci insiste alla fine del primo capitolo, non solo nell'imparare a memoria le diverse "porte" dell'addizione e della moltiplicazione, ma anche di esercitarsi lungamente al calcolo con la mente e con le mani cosa che certamente facilita l'agilità mentale e sviluppa le capacità cerebrali.

Fibonacci si rende conto di questa difficoltà e propone, all'inizio del terzo capitolo, un ulteriore algoritmo semplificato. Dice *«C'è infatti un altro modo di moltiplicare molto pregevole soprattutto per le moltiplicazioni di grandi numeri»* **(III.2)**. Questo algoritmo, a parte la disposizione delle cifre proposta da Fibonacci, è equivalente a quello in uso oggi. Il vantaggio è che i riporti sono in un qualche senso separati in due parti, ognuna delle quali di più facile memorizzazione. L'idea consiste nello sviluppare il prodotto

$$\boldsymbol{a\,x\,b} = (a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times (b_0 + b_1 10 + \ldots + b_{m-1}10^{m-1}) =$$
$$= (a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_0 + (a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_1 10 + \ldots +$$
$$+ (a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_{m-1}10^{m-1}$$

riducendolo alla somma di m prodotti del numero $\boldsymbol{a}$ per i numeri di una cifra b_0 , b_1, ..., b_{m-1} i cui risultati, essendo moltiplicati per potenze di 10, vanno opportunamente scalati. Ovviamente nell'eseguire il prodotto

$$(a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_0 = p_0 + p_1 10 + \ldots + p_n 10^n$$

si dovrà usare l'algoritmo precedente e quindi si dovrà tenere a memoria correttamente i vari riporti ma i numeri A_k da calcolare ora si riducono a un semplice prodotto di due cifre: $a_k b_0$. Questo prodotto va ora scritto in una opportuna tabella dopo di che si può vuotare la memoria, dimenticare i riporti precedenti e passare al prodotto successivo:

$$(a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_1 = q_0 + q_1 10 + \ldots + q_n 10^n$$

fino all'ultima cifra:

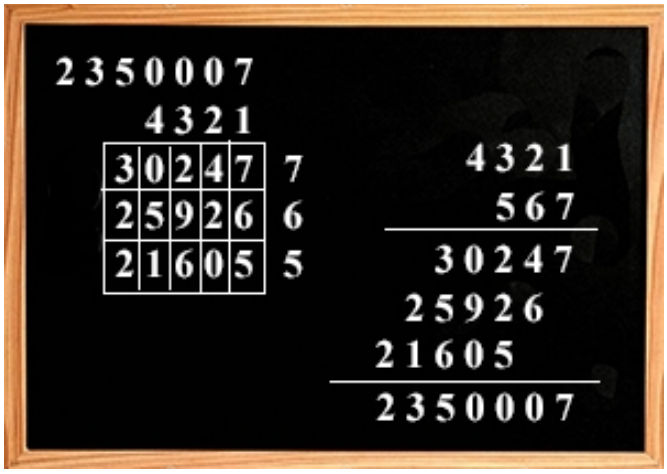
$$(a_0 + a_1 10 + \ldots + a_{n-1}10^{n-1}) \times b_{m-1} = s_0 + s_1 10 + \ldots + s_n 10^n$$

Fibonacci sceglie di scrivere in una tabella rettangolare i risultati dei prodotti parziali e due numeri $\boldsymbol{a}$ e $\boldsymbol{b}$ sui lati della tabella come in figura.

c_n	c_{n-1}	...	c_1	c_0	
	a_{n-1}	...	a_1	a_0	
p_n	p_{n-1}	...	p_1	p_0	b_0
q_n	q_{n-1}	...	q_1	q_0	b_1
...	...	...	...	...	...
s_n	s_{n-1}	...	s_1	s_0	b_{m-1}

Ora si devono sommare tutti i prodotti parziali posizione dopo posizione tenendo conto del fatto che tali prodotti sono moltiplicati per successive potenze di dieci. La cifra in prima posizione sarà dunque $c_0 = p_0$, quella in seconda $c_1 = p_1 + q_0$ e così via sommando le cifre in diagonale e trasferendo alla diagonale successiva un eventuale riporto. Il risultato viene segnato sopra il moltiplicatore *a*. Si vede che con questo procedimento occorre fare a memoria solo prodotto di cifre coi loro riporti, i prodotti calcolati vanno scritti nella tabella, e non tenuti a mente. In un secondo tempo si procede con la somma dei prodotti parziali lungo le diagonali tenendo in memoria eventuali riporti da trasferire nella diagonale successiva.

Torniamo ora alla lavagna ed eseguiamo il prodotto di 4321 per 567 mettendo a confronto il la disposizione di Fibonacci con quella attuale:



Vi sono altri tipi di moltiplicazione di origine indiana e cinese risalenti probabilmente al IV secolo e anche prima che richiedono ancora meno uso di memoria,

chiamati moltiplicazione a graticola o a gelosia, metodi, forse inconsapevolmente, ripresi e diffusi da Nepero nel 1617 probabilmente con l'intento di realizzare delle macchine moltiplicatrici.