

## NUMERI GRANDI DI FIBONACCI

come trovare velocemente i loro esatti valori numerici

**Sommario:** in questo articolo viene proposto, in alternativa al metodo classico per il calcolo dei Numeri della successione di Fibonacci, un algoritmo con il quale, tramite l'ausilio di un'altra importante successione di numeri, quella relativa ai Numeri di Lucas, è possibile calcolare il valore esatto di un numero grande di Fibonacci in un tempo molto più rapido. Per numeri di Fibonacci molto grandi composti da migliaia di cifre il tempo di calcolo risulta essere almeno di 35 volte più breve.

I numeri di Fibonacci sono sufficientemente noti per cui qui si accennerà solo alla loro definizione e si illustrerà un algoritmo per il calcolo rapido del loro esatto valore numerico. Per una descrizione dettagliata riguardante la loro implicazione in numerose e diverse discipline interessanti sia il campo della scienza o della tecnica o dell'economia come pure il campo dell'arte si rimanda alla lettura di numerosi e interessanti articoli comparsi sui Siti Internet.

I numeri di Fibonacci scaturiscono da una successione di numeri che obbediscono alle seguenti regole: partendo da due numeri iniziali che indicheremo con  $F(0)$  e  $F(1)$ , rispettivamente di valore 0 e 1 il successivo numero di Fibonacci  $F(2)$  è dato dalla somma di  $F(0)$  con  $F(1)$ , cioè  $F(2) = F(0) + F(1) = 0 + 1 = 1$ . I successivi numeri si trovano sommando sempre i due precedenti numeri. Pertanto per essi vale la formula seguente:  $F(k) = F(k-1) + F(k-2)$  dove con  $F(k)$  si indica il numero generico di Fibonacci che nella successione occupa il  $(k+1)$ -esimo posto con  $k$  denominato indice del numero.

Pertanto la successione generica di numeri di Fibonacci può indicarsi come segue:

$$F(0) = 0 \quad F(1) = 1$$

$$F(2) = F(0) + F(1) = 1$$

$$F(3) = F(1) + F(2) = 2$$

$$F(4) = F(3) + F(2) = 3$$

$$F(5) = F(4) + F(3) = 5$$

$$F(6) = F(5) + F(4) = 8$$

.....

$$F(k) = F(k-2) + F(k-1) = ?$$

.....

Ora, volendo conoscere il valore numerico di  $F(k)$ , se non si hanno disponibili per tali numeri altre formule risulta evidente che per trovarlo occorre conoscere e quindi calcolare tutti i precedenti numeri di Fibonacci. Questo metodo di procedere risulta adeguato solo se il valore dell'indice  $k$  non è molto grande. Ma se l'indice  $k$  ha valore grande, ad esempio  $k = 100000$  per trovare  $F(k)$  sarebbe necessario effettuare un numero di addizioni pari a  $k$  e cioè centomila di addizioni.

Realizzando su un Personal Computer in un qualsiasi linguaggio evoluto un semplice programma per il calcolo di  $F(100000)$  anche impiegando un PC sofisticato il tempo per effettuare 100000 operazioni aritmetiche di addizione non risulta indifferente (vedi Tabella in fondo all'articolo).

Sorge poi un altro problema: quanto può essere grande  $F(100000)$ ? Di quante cifre è composto?

Per rispondere a questo quesito fortunatamente la matematica mette a disposizione la seguente formula per il calcolo del numero di cifre<sup>(1)</sup>:

$$F(k) = \frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^k$$

Da tale formula uguagliandola a  $10^x$  si può trovare il valore dell'esponente  $x$  da cui si ricava immediatamente il numero  $N_c$  delle cifre componenti  $F(k)$  come *parte intera di  $x$  aumentata di 1*

si ha pertanto  $\frac{1}{\sqrt{5}} \left( \frac{1 + \sqrt{5}}{2} \right)^k = 10^x$  da cui  $x = k \log \left( \frac{1 + \sqrt{5}}{2} \right) - \frac{1}{2} \log 5$  e quindi  $N_c = \lfloor x \rfloor + 1$

Facciamo un esempio proprio con  $k=100000$ : svolgendo i calcoli relativi con la formula suddetta si trova  $x = 20898.41454...$  Quindi **20899** è il numero effettivo di cifre di cui risulta composto  $F(100000)$ .

---

(1) - per il calcolo del numero di cifre che compongono il numero di Fibonacci è lecito trascurare la parte con segno negativo della formula di Binet (vedi la formula a pag. seguente)

Ma pur conoscendo l'esatto numero di cifre di  $F(100000)$  come è possibile calcolare il suo valore esatto dato l'enorme numero di cifre implicate ?

Dobbiamo inevitabilmente fare ricorso ad operazioni aritmetiche che elaborano numeri molto grandi per i quali non risulta più sufficiente l'aritmetica che normalmente viene messa a disposizione sul PC dal software utilizzato.

Si presentano pertanto due problemi da risolvere:

il primo è quello di riuscire a manipolare e potere fare calcoli aritmetici esatti su numeri molto grandi; il secondo è quello di poter in qualche modo evitare di effettuare un numero grande di operazioni aritmetiche e cioè 100000 addizioni ad esempio per il calcolo esatto di  $F(100000)$

Per evitare di effettuare questo grande numero di addizioni si potrebbe pensare di ricorrere alla formula nota in matematica come formula di **Binet** [1],[2], che dà in forma concisa il valore di un qualsiasi numero di Fibonacci  $F(k)$ :

$$\frac{1}{\sqrt{5}} \left( \frac{1+\sqrt{5}}{2} \right)^k - \frac{1}{\sqrt{5}} \left( \frac{1-\sqrt{5}}{2} \right)^k$$

Sfortunatamente questa formula, nemmeno usando un'aritmetica a precisione multipla, si presta ad essere utilizzata per il calcolo esatto di  $F(k)$  in quanto, pur prescindendo dal grande numero di elevazione a potenza (difficoltà questa che si potrebbe facilmente superare) nella formula compare il numero irrazionale  $\sqrt{5}$ , per giunta anche a denominatore.

Per risolvere il primo problema come si è accennato, occorre fare ricorso all'uso di un'aritmetica a precisione multipla, messa a disposizione da specializzati pacchetti software o come si è fatto nei programmi realizzati creando opportune istruzioni nel linguaggio evoluto utilizzato.

Resta da risolvere il secondo problema.

Per esso bisogna fare ricorso all'impiego di un'altra famosa successione di numeri denominati *numeri di Lucas* prendendo in considerazione particolari relazioni che legano fra di loro i numeri delle due successioni.

Questa nuova successione, indicando con  $L(k)$  il suo generico termine è analoga alla successione di Fibonacci, ma con i suoi primi due termini aventi i seguenti valori.  $L(0) = 2$  e  $L(1) = 1$

Anche qui vale la relazione  $L(k) = L(k-2) + L(k-1)$

Pertanto i primi sette numeri della successione di Lucas sono i seguenti:

$$L(0) = 2 \quad L(1) = 1$$

$$L(2) = L(0) + L(1) = 3$$

$$L(3) = L(1) + L(2) = 4$$

$$L(4) = L(3) + L(2) = 7$$

$$L(5) = L(3) + L(4) = 11$$

$$L(6) = L(4) + L(5) = 18$$

.....

Come detto sopra, occorre considerare delle peculiari relazioni esistenti fra i numeri di queste due successioni, relazioni che si possono trovare in opportuni testi di matematica [3],[4],[5]. Da esse si possono ottenere dopo adeguate manipolazioni le seguenti finali relazioni che interessano il nostro caso, e che quindi prenderemo in considerazione:

per un indice  $k$  di *valore pari*, posto  $k = 2x$ : si ha :

$$F(k) = F(2x) = F(x) \cdot L(x) \tag{1p}$$

$$L(k) = L(2x) = L(x)^2 - 2 \cdot (-1)^x \tag{2p}$$

per un indice  $k$  di *valore dispari* posto  $k = 2x+1$  si ha :

$$2F(k) = F(2x+1) = F(x)L(x) + L(x)^2 - 2 \cdot (-1)^x \tag{1d}$$

$$2L(k) = L(2x+1) = L(x)^2 + 5F(x) \cdot L(x) - 2 \cdot (-1)^x \tag{2d}$$

Si fa osservare che per un indice  $k$  qualsiasi i valori di  $F(k)$  e di  $L(k)$  dipendono solo dall'indice  $x$  e dai

valori  $F(x)$  e  $L(x)$  dove risulta: per  $k$  pari  $x = \frac{k}{2}$  ; per  $k$  dispari  $x = \frac{k-1}{2}$

Consideriamo ora un qualsiasi indice  $k$ : posto  $k_n = k$  si costruisca a partire da  $k_n$  una successione di *indici interi* di valore numerico decrescente:

$$k_n, k_{n-1}, k_{n-2}, \dots, k_{i+1}, k_i, k_{i-1}, \dots, k_3, k_2, k_1$$

tale da rispettare le seguenti regole:

se il generico indice  $k_i$  è di valore pari si pone  $k_{i-1} = \frac{k_i}{2}$

se il generico indice  $k_i$  è di valore dispari si pone  $k_{i-1} = \frac{k_i - 1}{2}$

Si mette in evidenza che tale tipo di successione è limitata e termina sempre con il valore dell'indice  $k_1 = 1$ .

Essa risulta composta da un numero di indici uguale alla parte intera di  $\frac{\log k_n}{\log 2}$  aumentata di 1, dove con

il simbolo di  $\log$  si indica il logaritmo in base 10. Ad esempio per un indice  $k = 100000$  il numero totale degli indici è 17.

Partendo dall'indice  $k_1 = 1$  per il quale si ha  $F(k_1) = 1$  e  $L(k_1) = 1$  si possono calcolare in relazione ai vari indici della successione i valori di  $F(k_i)$  e di  $L(k_i)$  con l'impiego delle formule (1p), (2p), (1d), (2d) adattate come qui indicato:

per un  $k_i$  **pari**

$$F(k_i) = F(k_{i-1}) \cdot L(k_{i-1}) \tag{3p}$$

$$L(k_i) = L(k_{i-1})^2 - 2 \cdot (-1)^{k_{i-1}} \tag{4p}$$

per un  $k_i$  **dispari**

$$2F(k_i) = F(k_{i-1}) \cdot L(k_{i-1}) + L(k_{i-1})^2 - 2 \cdot (-1)^{k_{i-1}} \tag{3d}$$

$$2L(k_i) = L(k_{i-1})^2 + 5 F(k_{i-1}) \cdot L(k_{i-1}) - 2 \cdot (-1)^{k_{i-1}} \tag{4d}$$

si arriverà così ai valori cercati di  $F(k_n)$  e di  $L(k_n)$  con un ciclo limitato di iterazioni pari a  $\left\lceil \frac{\log k_n}{\log 2} \right\rceil$

vale a dire al massimo intero contenuto in  $\frac{\log k_n}{\log 2}$

Infatti con i valori di  $F(k_{i-1})$  e di  $L(k_{i-1})$ , si possono calcolare i valori di  $F(k_i)$  e di  $L(k_i)$  dove l'indice  $k_i$  se è pari è il doppio dell'indice  $k_{i-1}$ , se dispari è il doppio di  $k_{i-1}$  aumentato di 1. Si arriva così, partendo dai valori di  $F(1)$  e di  $L(1)$ , a trovare i valori di  $F(k_n)$  e di  $L(k_n)$  effettuando sulle formule mostrate nel riquadro il limitato suddetto ciclo di iterazioni.

Un semplice **esempio** chiarirà quanto detto.

Si voglia trovare il valore di  $F(k)$  e di  $L(k)$  per un indice  $k = 73$

Innanzitutto vediamo di quanti valori è composta la successione degli indici  $k_i$

Si ha  $\frac{\log 73}{\log 2} = 6,1898\dots$  e quindi il numero di iterazioni sarà pari a 6

La successione degli indici  $k$  tenendo conto delle regole summenzionate sarà poi la seguente:

$$k_7 = 73 \quad k_6 = 36 \quad k_5 = 18 \quad k_4 = 9 \quad k_3 = 4 \quad k_2 = 2 \quad k_1 = 1$$

utilizzando ora le relazioni (3p) e (4p) se  $k_i$  è pari e (3d) e (4d) se  $k_i$  è dispari. si ha :

1<sup>a</sup> iterazione:

$$F(2) = F(1) \cdot L(1) = 1$$

$$L(2) = L(1)^2 - 2(-1) = 1 + 2 = 3$$

2<sup>^</sup> iterazione:

$$F(4) = F(2) \cdot L(2) = 1 \cdot 3 = 3$$

$$L(4) = L(2)^2 - 2 \cdot (-1)^2 = 9 - 2 = 7$$

3<sup>^</sup> iterazione:

$$2 F(9) = F(4) \cdot L(4) + L(4)^2 - 2(-1)^4 = 3 \cdot 7 + 49 - 2 = 68 \text{ da cui } F(9) = 34$$

$$2 L(9) = L(4)^2 + 5 F(4) \cdot L(4) \cdot 2(-1)^4 = 49 + 5 \cdot 3 \cdot 7 - 2 = 152 \text{ da cui } L(9)=76$$

4<sup>^</sup> iterazione:

$$F(18) = F(9) \cdot L(9) = 34 \cdot 76 = 2584$$

$$L(18) = L(9)^2 - 2 \cdot (-1)^9 = 76^2 + 2 = 5778$$

5<sup>^</sup> iterazione:

$$F(36) = F(18) \cdot L(18) = 2584 \cdot 5778 = 14930352$$

$$L(36) = L(18)^2 - 2 \cdot (-1)^{18} = 5778^2 - 2 = 33385282$$

6<sup>^</sup> iterazione:

$$2 F(73) = F(36) \cdot L(36) + L(36)^2 - 2 \cdot (-1)^{36} = 149352 \cdot 33385282 + 33385282^2 - 2 = 1613045996450786$$

$$\text{da cui } F(73) = 806515533049393$$

$$2 L(73) = L(36)^2 + 5 \cdot F(36) \cdot L(36) - 2 \cdot (-1)^{36} = 33385282^2 + 5 \cdot 14930352 \cdot 33385282 - 2 = 606847113615842$$

$$\text{da cui } L(73) = 1803423556807921$$

Da una prima analisi di tale algoritmo potrebbe apparire che esso sia più complicato e addirittura più lento nei tempi di calcolo rispetto a quello classico che si avvale delle formule ricorsive  $F(k) = F(k - 1) + F(k-2)$  per calcolare i numeri di Fibonacci o  $L(k) = L(k-2) + L(k-1)$  per calcolare i numeri di Lucas

In effetti, se ci si limita al calcolo dei primi numeri di queste successioni si può realizzare con le suddette formule, impiegandole in modo iterativo, un programma molto semplice e con tempi di calcolo per numeri di Fibonacci  $F(k)$  (o di Lucas) con indici  $k$  non superiori a 500 paragonabili ai tempi di calcolo ottenuti con il presente algoritmo.

Ma se consideriamo indici più grandi ad esempio per  $k > 5000$  i tempi di calcolo per trovare il valore del numero di Fibonacci (o di Lucas) con i due algoritmi risultano considerevolmente diversi.

Per confrontare fra loro i due algoritmi, quello classico e quello illustrato in questo articolo, si sono approntati sullo stesso PC due programmi<sup>(2)</sup> entrambi scritti nello stesso linguaggio evoluto (nel nostro caso in Qbasic), nei quali viene utilizzata per le istruzioni riguardanti le operazioni aritmetiche lo stesso tipo di aritmetica a precisione multipla, necessaria in quanto si vogliono avere i valori esatti per i numeri di Fibonacci (o di Lucas) composti da molte cifre (anche decine di migliaia).

Ebbene, dalla **Tabella** sotto riportata è evidente la notevole efficienza dell'algoritmo illustrato rispetto all'algoritmo classico nei riguardi del tempo necessario per il calcolo del *numero di Fibonacci*  $F(k)$ .

In essa con **Tcc** si è indicato il tempo di calcolo relativo all'algoritmo classico, con **Tcp** quello relativo al presente algoritmo., con **Nc** il numero di cifre che compongono  $F(k)$  o  $L(k)$ .

Il tempo di calcolo **Tcp** è comprensivo anche del calcolo del *numero di Lucas*  $L(k)$ .

<b>F(k)</b>	<b>Tcc</b> (secondi)	<b>Tcp</b> (secondi)	<b>Nc</b>
<b>F(5000)</b>	<b>1.43</b>	<b>0.06</b>	<b>1045</b>
<b>F(10000)</b>	<b>5.38</b>	<b>0.25</b>	<b>2090</b>
<b>F(15000)</b>	<b>10.93</b>	<b>0.44</b>	<b>3156</b>
<b>F(20000)</b>	<b>19.06</b>	<b>0.76</b>	<b>4180</b>
<b>F(40000)</b>	<b>74.54</b>	<b>1.95</b>	<b>8360</b>
<b>F(50000)</b>	<b>115.73</b>	<b>3.08</b>	<b>10450</b>
<b>F(100000)</b>	<b>451.65</b>	<b>11.37</b>	<b>20899</b>
<b>F(200000)</b>	<b>1829.79</b>	<b>43.18</b>	<b>41798</b>

(2) – Non si è ritenuto opportuno riportare i listati dei due programmi in Qbasic per non appesantire troppo l'articolo

## CONCLUSIONI

Con l'algoritmo illustrato il tempo di calcolo per trovare, non solo il valore esatto di un numero grande di Fibonacci, ma contemporaneamente anche l'analogo numero grande di Lucas, risulta molto minore di quello impiegato con l'algoritmo classico, come si evince chiaramente dai dati riportati nella **Tabella**. La differenza di tempi è facilmente spiegabile se si considera che con l'algoritmo classico per trovare il valore esatto del numero in esame occorre effettuare pedissequamente il calcolo di tutti i precedenti numeri di Fibonacci, impiegando un insieme molto elevato di cicli iterativi, mentre con l'algoritmo illustrato è sufficiente, pur con calcoli aritmetici più complessi in ogni iterazione, un numero molto limitato di cicli.

Si riporta qui di seguito un esempio di risultato trovato con l'algoritmo illustrato:

### VALORE ESATTO DEL NUMERO DI FIBONACCI $F(15100)$

```
F(15100) = 2311440787374354574679920870523086410578257933726209923286764174865
40378558039538434890384479456861448991375905881999358048370129113460964103555573
85002169295039957333172585494782340375222432683062031952728334827407135862698213
31772538391478709946780546940633373925218073637056767122460051795667168373017437
23961368200476908802510561466527563831138656674603358444662373012886883686108988
46082235328726180496367311107627545455464202308808416313654627541956150139285482
84835028560747778488872936778927895697784695782119990760732062543295402873837424
32654841405330452998801793882316704963945307897377628177188570309250409256523912
19393306591388244297275464780769883730925169708733509936715451455574548706190736
81788013078160449002837445419675238753713425146289241268613196701031315502139912
88518323822713817673590274016202837777306639296890579347341749894060972687234208
42080045693528037254976474607318238677338011319100220544689721943930703740132076
78049738956404860835201023528564938856515609049203855542927073524833524257207555
62201513066177950661390889784758766302067802317404673414014809958651157354707149
12724134569989118967584212504962579730389458930863899084098430479430993885715031
21304199720827738649545181412691446320251584459772515681888843251445513615013512
10108276863183454166836127470188078717416485617750773693957382418434084745346020
12651758124605699565095626376043695895654889715001188613486586987189259524700728
18597900920768356939302450245949624073269871297324372413304254517745183659468685
21147224199287930124970800956317659374775986835865484905361563892502058704238410
69233311288832524926297715853758598508047197655545598299061322336473668655549777
24923421527868807940464689269275203891262270789418913599203360687514303159199284
93190718620584429194925284323215443382276209769991882164120472345944617913954870
68641319929125047436915286291707692021004238899457872126036340419952896962349498
41242098264870125735955019117616533181044513880304191387918401938521457033233389
78319138880533960464130865540752697274836037420960740312174789193363004974902775
73187400503636383483305547867782111791119929370956648818247151638734375561664843
34890274667125139442294818300675756027858215059458789489597319575615084613700802
36750724406693255646704397772650723291894932136279224107876092241177347806172634
90282736390225668134178275304513827602090947503636130125678452885157929429502480
18234695862766012493741892505319190263897544698141535021715122555381651078483566
29134541855473930329260091971519102688468393601226897230716096675372428283443088
55821427853154476690247903389283548059050774366991663248551511071773469996512398
85880215113842873374859856832076129199269067724073720837012730035509764102157103
96606361219106568688450478823901859488850323320915282936567834623934184065181325
45626283203470747429291692004500763818092873283803617284392605430704348020652941
64398965070242607054476668252840823697004092588378835482254310447580512059287740
46894789690288433726004757132431678156074252508968008079715122559427986944592932
04793580636188829777452434659694086610055031733940142255376540401867920230277705
6990077403597953537195834911702710463079342145075
```

**$F(15100)$  e' composto da 3156 cifre**

## VALORE ESATTO del NUMERO di LUCAS (15100 )

L( 15100 ) = 5168538726534694464874836101571198178970085281716709754527080341622  
04246855136057224768766605789560332500750815987549336506992454686769829527721512  
23932919317896324393105821646293924697214342669537378547559574034502951214307794  
92685352979335856921716807659416801494358957276078908543328940436812468020498580  
48878853571268446846457247997040530437683128160621463584525026206468976672909525  
47505562438428804487531196860956866458909814210079086945634025921009058245558192  
33418346428026354412440488246129145158932842090215629555009179292567271613281204  
04143045832121660252907204747449581784924640861065359974550593900439673832282522  
88197132471159736684670595908544811313180196490447319488937831047885601981537101  
49091553731808711079502922503968399187604038489338069726919417115220511405186374  
74613290366184288321971091329147722203375335668828304319572221356435643359003786  
56757330543505970735017528210766820134969536893010130319896521851336218945491113  
27704095414530203661161410780303535307249134870827713097075228909372992320351821  
58859016100891650874253464778244448674234644166060937761487638032559174600880654  
05665705529772767067933544956732212921425386729971913218166202090493752858937850  
86406764788366035486772536237131065533730806918724965206603113954927686726931349  
87808695615820993594865936548490077401276655025128279805884298376718501319914129  
30515363947015401272066261144921027762821935837581188445191443469753328448138760  
41249110048902825924925489067155911333553103228456907167592469121158260397088368  
19399324461556683171620595230001419651094411367647929762314329596515698555511987  
23945344198409448012579413392959567946980879976390470745289650305217745977127529  
23618887746042480579656854454987839643614231810816933987428958608278352015237979  
03380018359589055085664504563730942747903203113244706251194928853893267082261776  
56534195905234845706930937906899475652273788975981852244136497543027023229391887  
32405119147159063736533553167391312591353648884411990566335902830197524271668216  
52536101375297387323043034838629011571082443176008399026519285885602748374385839  
29378178616198901341831039684097840951926838180696957185924718772914626715054562  
20305459030775150244659978367841886805387302385597536918146258940081120006383794  
41742644055352935953071674050335900432494874375885477873357146619910452034132309  
09597088083620035608477023773692875650191800027269245060251504813288561026822845  
61967030845972548081040951586235679118317005805200168286071333879150667817905654  
68723538785032048113819715667714921631239462759138558971983582096675792299672758  
08722845551782084248475578579548670581837720636239609918376150176345936622516983  
32629554219129227091155236784647812837387354159960070845846051624922658522004453  
50510919255305659797401981452673920669441776504332889161871819789818312136683318  
89727689205757062602773721020160961018385328856629119443760432526198733306764740  
15208800957268711504756522498216397860696572652873782662864445155399928284092715  
88961419437931128344024166946393224677305599716044909318404060410896364195151632  
75839769214569319552644496024000485246424955618450484808142804818527044218839895  
6672299369678476947784896478262364045337306003127

**L( 15100 ) e' composto da 3156 cifre**

**Il tempo totale impiegato per il calcolo di F(15100) e di L(15100) è risultato di 0.45 secondi**

### RIFERIMENTI

[1] [http://it.wikipedia.org/wiki/Jacques\\_Binet](http://it.wikipedia.org/wiki/Jacques_Binet)

[2] <http://www.matematicamente.it/cimolin/formula/formula6.htm>

[3] P.Ribenboim, *The Book of Prime Number Records* , Ch.2 - 2nd ed.–Springer-Verlag ,1989

[4] D.M. Bressoud, *Factorization and Primality Testing*, Ch.12 – Springer -Verlag ,1989

[5] H. Riesel, *Prime Numbers and Computer Methods for Factorization* ,Ch.4 - 2nd ed. – Birkhäuser 1994