Le Terne Pitagoriche particolari

(composte da centinaia di cifre…)

di Giovanni Di Maria
email: calimero22@freemail.it

Di **terne pitagoriche** si è parlato già ampiamente, anche sulla rete: si tratta di un gruppetto formato da tre numeri (rispettivamente i due cateti e l'ipotenusa di un ipotetico triangolo rettangolo) obbligatoriamente interi e che soddisfino l'eguaglianza:

$$c1^2 + c2^2 = ipo^2$$

dove *c1* e *c2* costituiscono i due cateti e *ipo* costituisce l'ipotenusa.

Per essere una Terna Pitagorica occorre quindi che si verifichi la condizione generale, ossia che i tre parametri siano tutti numeri interi.

Esistono naturalmente infinite combinazioni possibili, in quanto si tratta sempre di una formazione non critica ad alta probabilità di avvenimento.

Esempi importanti (semplici da calcolare anche mentalmente) sono rappresentati dai gruppi (3, 4, 5) oppure (5, 12, 13).

In questo articolo tratterò solamente di quelle terne in cui la misura di un cateto sia consecutiva (quindi distante di una unità) all'altro cateto.

Iniziamo subito da un esempio, poi fornirò un metodo *balistico* per il calcolo e la determinazione.

Le terne (3, 4, 5) e (20, 21, 29) e (119, 120, 169) e (696, 697, 985) e così via, costituiscono validamente delle terne pitagoriche in quanto, come si può notare, ciascun cateto dista dall'altro cateto proprio di una unità, ed inoltre i tre parametri sono numeri interi.

Provate ad effettuare, con il Teorema di Pitagora, il calcolo e vedrete che i gruppi sono verificati.

Calcolare le terne con l'ausilio di un elaboratore è oltremodo semplice. Utilizzando un linguaggio di programmazione è possibile fare in modo che una variabile incrementi il suo valore di una unità sino alla ricerca del risultato voluto.

Ma cosa succede quando la "fame" di numeri e di risultati aumenta a dismisura sino a pretendere quantità di lunghezza spropositata e di grandezza stratosferica?

Con i normali metodi di calcolo possiamo sicuramente abbandonare il nostro lavoro per la presenza di due problemi apparentemente irrisolvibili:

1) I normali linguaggi di programmazione basano i propri valori numerici su pochi bit, pertanto la precisione e l'accuratezza (quindi il numero delle cifre composte) è estremamente basso. In altre parole potremo solo usare numeri formati al massimo da 8-12 cifre significative e "realmente" esistenti, in quanto la semplice notazione esponenziale non si presta affatto per lo scopo;
2) Ammesso che il linguaggio di programmazione ci permetta di risolvere il problema n. 1, ci troviamo davanti "all'insormontabile" fattore **tempo**, che purtroppo ci mette davanti un limite pratico di calcolo. Basti pensare infatti che, per calcolare una terna consecutiva composta dai numeri (927538920, 927538921, 1311738121) occorrerebbero, in media, 30 giorni. Questo tempo proibitivo si avrebbe però utilizzando un algoritmo non ottimizzato, quindi non idoneo.

Vediamo allora come generare delle Terne Pitagoriche, di cateti consecutivi, in un tempo pressoché immediato.

Abbandoniamo a priori l'utilizzo di un semplice linguaggio di programmazione in quanto, come detto, non dispone della dovuta precisione.
Utilizziamo pertanto l'interprete Basic **UBASIC**, liberamente distribuito sulla rete ed operante in Dos. Questo interprete, quasi uguale al cugino GWBASIC, permette una precisione straordinaria nella gestione dei numeri arrivando, a volte, a gestire bene numeri sino a 2000 cifre !!!
Analizziamo passo dopo passo il problema. Fornirò di volta in volta vari algoritmi, sempre più perfezionati.

ALGORITMO (1° metodo estremamente LENTO)

Per trovare le terne abbiamo bisogno di "processare", attraverso un contatore, tutti i numeri progressivi da 1 a *limite*, quest'ultimo sufficientemente grande.
Per ogni numero quindi si effettuano le seguenti operazioni:

- Il valore del contatore costituisce la misura di un cateto (C1);
- Calcolare la misura dell'altro cateto semplicemente aggiungendo una unità, per determinare la consecutività (C2);
- Determinare con il Teorema di Pitagora la misura dell'ipotenusa (IP);
- Controllare se tale ipotenusa (IP) sia un intero o meno;
- Se IP è intero allora possiamo stampare i valori di C1, C2 e IP e possiamo considerarli membri di una Terna Pitagorica di cateto consecutivo.

L'idea è effettivamente buona, se non fosse per il tempo che ostacola e ci mette il bastone tra le ruote. Come detto prima, per arrivare a processare tutti i cateti da 1 a un miliardo occorre circa un mese di tempo, gestendo solamente 9 cifre. Quanti anni occorrerebbero per processare cateti formati da centinaia di cifre? Non basterebbero sicuramente migliaia di vite dell'universo e nemmeno un super computer in grado di elaborare miliardi di istruzioni in un pico-secondo, per portare a termine il lavoro.

Ma esiste una soluzione. Propongo innanzitutto il listato in UBASIC, per la ricerca delle terne. Il metodo utilizzato pertanto (per il momento) è quello della "*forza bruta*" in quanto esso cerca indiscriminatamente tutti i possibili valori da 1 a *limite* (nell'esempio 10^200).

```
10   point 200
20   cls
30   N=0
40   while N<10^200
50         N=N+1
60         C1=N:C2=N+1:IP=sqrt(C1*C1+C2*C2)
70         R=IP-int(IP)
80         if R=0 then print C1;C2;IP
90   wend
```

Il listato è funzionante ma non funzionale. E' estremamente lento. Commentiamo le varie istruzioni presenti nei numeri di linea:

10 Predispone la gestione sino a 963 decimali;
20 Cancella il video;
30 Inizializza il contatore a zero (punto di partenza);
40 Crea un ciclo ripetitivo che si ripeterà sino a quando N arriva a 10E200 !!!
50 Incrementa di una unità il contatore;
60 Assegna a C1 il valore di N, C2 il valore successivo e calcola ipotenusa;
70 Controlla se ipotenusa è intero, guardando se ha cifre decimali;
80 Se è intero allora stampa la TERNA PITAGORICA;
90 Fine ciclo.

Come si vede il listato è semplice.

Riporto adesso alcuni risultati generati dopo alcune ORE di funzionamento:

CATETO 1	CATETO 2	IPOTENUSA		Rapporto di Cateto1 e Cateto1 precedente
3	4	5		
20	21	29		6,666666667
119	120	169		5,950000000
696	697	985		5,848739496
4.059	4.060	5.741		5,831896552
23.660	23.661	33.461		5,829021927
137.903	137.904	195.025		5,828529163
803.760	803.761	1.136.689		5,828444631
4.684.659	4.684.660	6.625.109		5,828430128
27.304.196	27.304.197	38.613.965		5,828427640

Siamo ancora ben lungi dal parlare di GRANDI NUMERI.
Dopo un'attenta osservazione alle terne generate, salta subito all'occhio un particolare veramente importante: guardate il *rapporto* che esiste tra un qualsiasi cateto1 ed il cateto1 precedente (esempio: 119/20 oppure 23.660/4.059 oppure ancora 4.684.659/803.760).

Come si può ben osservare esso tende, sempre di più, al numero 5,82842712475 cioè:

$$5{,}82842712475 = 3 + \left(2\sqrt{2}\right)$$

Questo vuol dire che se, appena viene trovato una terna, moltiplichiamo il cateto 1 per tale coefficiente, "saltiamo" direttamente ad esaminare il prossimo cateto, con il risparmio di tantissimo tempo.

ALGORITMO (2° metodo estremamente VELOCE)

Sviluppiamo pertanto l'algoritmo con questa nuova ottimizzazione, che consentirà di trovare rapidamente le terne pitagoriche, formate anche da centinaia di numeri, senza dover attendere miliardi di anni ….

Per ogni numero quindi si effettuano le seguenti operazioni:

- Il valore del contatore costituisce la misura di un cateto (C1);

- Calcolare la misura dell'altro cateto semplicemente aggiungendo una unità, per causare la consecutività (C2);
- Determinare con il Teorema di Pitagora la misura dell'ipotenusa (IP);
- Controllare se tale ipotenusa (IP) sia un intero o meno;
- Se IP è intero allora possiamo stampare i valori di C1, C2 e IP e possiamo considerarli una Terna Pitagorica di cateto consecutivo, ma a differenza del precedente algoritmo, il contatore adesso è **moltiplicato** per la costante 5,82842712475, in modo che esso stesso avanzi più "velocemente" rispetto l'incremento di una sola unità ma soprattutto che vengano "saltate" le altre misure dei cateti che non fornirebbero sicuramente risultati utili.

In questo modo, in pochissimi istanti verranno visualizzati sullo schermo tutte le terne pitagoriche formate da centinaia di cifre, sino al limite impostato a 10^200. Ecco il listato definitivo:

```
10  point 200
20  cls
30  Rapp=sqrt(2)*2+3
40  N=0
50  while N<10^200
60        N=N+1
70        C1=N:C2=N+1:IP=sqrt(C1*C1+C2*C2)
80        R=IP-int(IP)
90        if R=0 then:print C1:print C2:print IP:print:N=int(N*Rapp)
100 wend
```

Commentiamo le righe di programma:

10 Definisce la precisione numerica a 963 cifre decimali;
20 Cancella il video;
30 Crea la variabile *rapp* contenente il coefficiente di incremento geometrico;
40 Inizializza a zero il contatore;
50 Crea un ciclo ripetitivo che si ripeterà sino a quando N arriva a 10E200 !!!
60 Incrementa di una unità il contatore;
70 Assegna a C1 il valore di N, C2 il valore successivo e calcola ipotenusa;
80 Controlla se ipotenusa è intero, guardando se ha cifre decimali;
90 Se è intero allora stampa la TERNA PITAGORICA e in più <u>innalza il valore</u> del contatore al prossimo della serie del cateto successivo;
100 Fine ciclo

Ecco la tabella dei numeri generati. Ho dovuto troncarla sino a questo punto perché lo spazio occupato dai numeri è cresciuto rapidamente. Ma vi assicuro che le cifre generate sono tantissime.

Cateto 1	Cateto 2	Ipotenusa
3	4	5.0
20	21	29.0
119	120	169.0
696	697	985.0
4059	4060	5741.0
23660	23661	33461.0
137903	137904	195025.0
803760	803761	1136689.0
4684659	4684660	6625109.0
27304196	27304197	38613965.0
159140519	159140520	225058681.0
927538920	927538921	1311738121.0
5406093003	5406093004	7645370045.0
31509019100	31509019101	44560482149.0
183648021599	183648021600	259717522849.0
1070379110496	1070379110497	1513744654945.0
6238626641379	6238626641380	8822750406821.0
36361380737780	36361380737781	51422757785981.0
211929657785303	211929657785304	299713796309065.0
1235216565974040	1235216565974041	1746860020068409.0
7199369738058939	7199369738058940	10181446324101389.0
41961001862379596	41961001862379597	59341817924539925.0
244566641436218639	244566641436218640	345869461223138161.0
1425438846754932240	1425438846754932241	2015874949414289041.0
8308066439093374803	8308066439093374804	11749380235262596085.0
48422959787805316580	48422959787805316581	68480406462161287469.0
282229692287738524679	282229692287738524680	399133058537705128729.0
1644955193938625831496	1644955193938625831497	2326317944764069484905.0
9587501471344016464299	9587501471344016464300	13558774610046711780701.0
55880053634125472954300	55880053634125472954301	79026329715516201199301.0
325692820333408821261503	325692820333408821261504	460599203683050495415105.0
1898276868366327454614720	1898276868366327454614721	2684568892382786771291329.0
11063968389864555906426819	11063968389864555906426820	15646814150613670132332869.0
64485533470821007983946196	64485533470821007983946197	91196316011299234022705885.0
375849232435061491997250359	375849232435061491997250360	531531081917181734003902441.0
2190609861139547943999555960	2190609861139547943999555961	3097990175491791170000708761.0
12767809934402226172000085403	12767809934402226172000085404	18056409971033565286000350125.0
74416249745273809088000956460	74416249745273809088000956461	105240469650709600546001391989.0
433729688537240628356005653359	433729688537240628356005653360	613386407933224037990008001809.0
2527961881478169961048032963696	2527961881478169961048032963697	3575077977948634627394046618865.0
14734041600331779137932192128819	14734041600331779137932192128820	20837081459758583726374271711381.0
…..	…..	…..
…..	…..	…..

Riporto infine, solo per curiosità, l'ultima terna generata dopo pochissimi istanti: si tratta di un numero enorme:

Cateto1:
38.870.796.548.368.940.451.592.529.482.185.869.982.938.448.205.812.640
.195.914.560.739.542.103.841.403.178.847.163.517.462.769.143.179.065.0
31.576.973.812.014.377.488.506.777.895.445.800.461.891.869.308.645.201
.761.858.965.032.907.136.032.847.098.509.289.762.520.539

Cateto2:
38.870.796.548.368.940.451.592.529.482.185.869.982.938.448.205.812.640
.195.914.560.739.542.103.841.403.178.847.163.517.462.769.143.179.065.0
31.576.973.812.014.377.488.506.777.895.445.800.461.891.869.308.645.201
.761.858.965.032.907.136.032.847.098.509.289.762.520.540

Ipotenusa:
54.971.607.658.948.646.301.386.783.144.964.782.698.772.613.513.307.493
.180.078.896.702.918.825.851.648.683.235.325.858.118.170.150.873.214.9
78.343.601.463.118.106.546.653.220.435.805.362.395.962.991.295.556.488
.036.606.954.237.309.847.762.149.971.207.793.263.738.989

Come si può notare, il cateto1 e il cateto2 differiscono di una unità, quindi sono consecutivi.
Neanche le stelle dell'Universo arrivano a simili quantità!

Bene, con questo è tutto. Spero che vi siate appassionati all'argomento e vi do appuntamento per i prossimi articoli

Giovanni Di Maria