

Block Notes Matematico

Frazioni: primalità e fattorizzazione

Rosario Turco, Maria Colonnese

Con le frazioni è possibile sia verificare tematiche di primalità che di fattorizzazione, entrambi passanti attraverso la determinazione del periodo di una frazione.

In questo articolo gli autori si pongono, quindi, il problema di scrivere un algoritmo per determinare il periodo di una frazione.

Il risultato di una frazione può dare:

- assenza di periodo: numero finito o numero irrazionale
- presenza di periodo
- presenza di antiperiodo e periodo

Se al denominatore ci sono 2 e 5 o potenze di essi o combinazioni (perché fattori della numerazione per 10), avremo a che fare con un numero finito.

Si definisce **gaussiano del numero N rispetto ad una base B**, indicato come $k = g(N) = T(1/N)$, "il più piccolo valore k tale che $B^k = 1 \pmod{N}$ ".

Si dimostra che il gaussiano k di un rapporto a/b dipende solo dal denominatore b, quindi nel calcolo del gaussiano si può porre a=1 (come nell'algoritmo proposto). Ecco perché spesso è indicato come $k = T(1/N)$.

Un teorema che deriva dal Piccolo Teorema di Fermat porta a dire anche che il gaussiano di n è un divisore della funzione di Eulero $\varphi(n)$. In particolare se n è primo è $\varphi(n) = n - 1$.

Un algoritmo del gaussiano è presentato in APPENDICE.

Primalità? Meglio la compositezza.

Con le frazioni continue e l'espansione periodica in base 10 è possibile creare un "test di compositezza" ad hoc.

E' brutto il termine "compositezza", ma l'italiano non offre di meglio, anche perché "compostezza" ha un altro significato!.

Teorema (R. Turco)

"Sia $n > 1$ e dispari. Se il quoziente di $1/n$ ha un gaussiano $k = T(1/N)$ diverso da 1 e $(n-1)/T$ dà un quoziente non intero, allora n

non è primo. Se il gaussiano di $1/n$ equivale a $n-1$, allora n è primo".

Esempi

$T(1/57)=18$ (T diverso da 1)
 $(57-1)/18=3.1111$ non intero
allora 57 è composto.

$T(1/3337)=1610$ (T diverso da 1)
 $(3337-1)/1610=2.0720496$ etc
allora 3337 è composto.

$T(1/3333)=4$ con $(3333-1)/3=883.0$
allora il metodo non può dire se è composto.

$T(1/9)=1$ **violazione della regola**
 $(9-1)/1=8$ intero il metodo non può dire se è composto.

$T(1/3367)=6$ $(3367-1)/6$ il metodo non può dire se è composto.

$T(1/3343)=3342=\varphi(3343)$ è primo!!!

Valutazione del metodo

Se si guarda l'algoritmo `isComposite` ci si rende conto che, per numeri grandi, già ad esempio per un numero di Mersenne come $2^{31}-1$, poiché il gaussiano è calcolato per successivi incrementi unitari ne consegue che sono necessari molti cicli.

In tal caso è più veloce un algoritmo come `isMyPrime` o le funzionalità built-in di PARI/GP come `eulerphi`, `isprime`, `ispseudoprime`. Queste ultime funzioni built-in soffrono ovviamente del fatto che dipendono dal numero di primi precaricati e dalla RAM disponibile e allocata al programma.

Fattorizzazione dei semiprimi

Tra le fattorizzazioni più curiose e simpatiche di un semiprimo $N=p*q$ c'è la tecnica della "Espansione periodica in base 10 della frazione $1/N$ ".

Facciamo un esempio: $N=1517$. Occorrono tre step per la fattorizzazione del semiprimo.

Step 1: trovare la lunghezza del periodo $T(1/N)$ della frazione $1/N$

$$1/N=0.\mathbf{000659195781147}000659195781147$$

dove per semplicità abbiamo marcato in bold rosso solo la "parte periodica" della frazione $1/N$. Se contiamo le cifre di tale periodo (bold rosso) si ottiene che $T(1/N)=15$.

Step 2: fattorizzare la lunghezza del periodo della frazione $1/N$

Si fattorizza un numero inferiore di quello di partenza; in particolare fattorizziamo adesso la lunghezza del periodo $T(1/N) = 15 = 3 \cdot 5$. Stesso risultato è ottenibile da PARI/GP con `factor(15)`, dove $k_1=3$ e $k_2=5$.

Se il periodo è costituito dal prodotto di più di 2 fattori, occorre "combinare le partizioni dei fattori", cosa che vedremo in seguito; mentre ora proseguiamo con l'esempio $N=1517$.

Step 3: trovare il MCD (in inglese GCD) di N con 10^{k_1-1} e di N con 10^{k_2-1}

$$\begin{aligned} \text{GCD}(1517, 10^3-1) &= 37=p \\ \text{GCD}(1517, 10^5-1) &= 41=q \end{aligned}$$

Allora $N=1517=37 \cdot 41$.

Controprova

Facciamo la contro-prova del metodo di sopra. Sappiamo che $N=1517=37 \cdot 41$.

Ora cerchiamo per $p=37$ e $q=41$ i più piccoli valori k_1 e k_2 tali che siano intere le quantità:

$$\begin{aligned} (10^{k_1}-1)/p &= 3 \\ (10^{k_2}-1)/q &= 5 \end{aligned}$$

Cosa sono allora k_1 e k_2 ? Sono la lunghezza dei periodi delle espansioni di $1/p$ e $1/q$.

Come si dimostra che il tutto è vero? Intanto ricordiamo che lcm è il minimo comune multiplo (mcm), mentre LCM è il prodotto dei minimi comuni multipli in gioco; inoltre indichiamo con T il periodo di una frazione.

Se $N=p \cdot q$ allora è: $T(N)=LCM(T(p), T(q))$

Se indichiamo $g = \text{GCD}[T(p), T(q)]$ allora $T(N) = T(p)T(q) / g$.

Così per qualche fattorizzazione $T(N) = abg$ noi avremo:

$$T(p) = ag \text{ e } T(q) = bg$$

ed in conclusione:

$$\begin{aligned} p &= \text{GCD} [N, 10^{ag} - 1] \\ q &= \text{GCD} [N, 10^{bg} - 1] \end{aligned}$$

In tutto questo abbiamo usato la base $B=10$ ma nella dimostrazione al posto del 10 potevamo mettere genericamente B :

$$p = \text{GCD} [N, B^{ag} - 1]$$

$$q = \text{GCD} [N, B^{bg} - 1]$$

Simpatico no? E' un metodo che si applica bene e facilmente ad un numero semiprimo; mentre le cose si complicano di più se non è semiprimo o se il periodo non si riesce a individuare.

Se N è semiprimo ad esempio in PARI/GP possiamo saperlo con la funzione bigomega(N): se N è semiprimo difatti essa restituisce 2.

La bigomega fornisce il numero di fattori primi anche se ripetuti. In verità dopo qualche prova si comprende che il metodo vale la pena applicarlo per semiprimi N di valore grande.

Difficoltà superabili

1. In PARI/GP non esiste una funzione predefinita (built-in) che individua il periodo: occorre scrivere un algoritmo per $T(1/N)$;
2. Non sempre la frazione è periodica;
3. Per N grande, PARI/GP restituisce il valore in notazione esponenziale (esempio di notazione esponenziale: 23 E-21)

Per il punto 1 cercare il periodo di una frazione significa cercare "il più piccolo valore k tale che $B^k = 1 \pmod N$. Scrivere un algoritmo che calcolo il gaussiano comunque è semplice, ma sicuramente il metodo non è veloce per numeri grandi.

In PARI/GP esiste anche un ulteriore metodo semplice da collocare comunque nell'ambito di un algoritmo; ad esempio sappiamo che $\text{sqrt}(21) = 4, \underline{1,1,2,1,8}$ dove la parte sottolineata è il periodo $T(\text{sqrt}(21))$.

Se usiamo $\text{contfrac}(\text{sqrt}(21))$ otteniamo il vettore

$$[4, 1, 1, 2, 1, 8, 1, 1, 2, 1, 8, 1, \dots]$$

Ora se a_1 è il primo elemento del vettore e a_j è il j-esimo elemento del vettore, allora il periodo $T(\text{sqrt}(21)) = j-1$ se $a_j = 2 \cdot a_1$. Ovviamente il metodo è poco applicabile se $a_1 = 0$, ovvero la frazione non è periodica ma è finita o infinita (numero irrazionale).

L'algoritmo per il calcolo del gaussiano presente in APPENDICE, non usa però contfrac . La scelta dipende dal fatto che contfrac non sempre fornisce il risultato in modo facilmente trattabile nei casi di rapporti $1/N$. Contfrac è più efficace nei casi di calcoli delle radici quadrate di un numero N.

Vediamo qualche altro esempio un po' più complicato, per comprenderne la gestione, comunque non semplice. L'esempio di fattorizzazione visto precedente era un caso banale.

Esaminiamo $N = 66167$ $T(66167)=1092$. La fattorizzazione di $1092 = 2*2*3*7*13$.

Qui si deve vedere come combinare le possibili partizioni di 1092.

Ad esempio dopo qualche tentativo si vede $1092=21*26*2$ da cui $ag=42$ e $bg=52$. In questo caso $T(p)$ e $T(q)$ non sono coprimi.

Da qui:

$$\begin{aligned} \text{GCD}[66167,10^{(21*2)}-1] &= 127 \\ \text{GCD}[66167,10^{(26*2)}-1] &= 521 \end{aligned}$$

Ovviamente basta calcolarne solo uno dei fattori, l'altro è ottenibile per divisione, finché non si ottiene un fattore non banale tale che $\text{GCD}[N,B^k-1]$.

Ad esempio nel caso $N=57$ $T(1/N)=18=2*3*3$
Un partizionamento soltanto è: $2*3$

$$\text{GCD}[57,10^{(6)}-1] = 3$$

Ora l'altro fattore è $57/3 = 19$

Questo tipo di fattorizzazione può lavorare su numeri inferiori rispetto a N , ed è utile soprattutto per i semiprimi. Ma, come visto, non è detto che sia più veloce di una fattorizzazione di tipo trial, a causa del gaussiano da trovare.

Cosa giustifica che il metodo delle frazioni continue sia corretto per fattorizzare un numero RSA o semiprimo?

Abbiamo visto in [8] che un metodo per fattorizzare un numero RSA è di usare un'equazione di secondo grado: difatti sia il prodotto delle soluzioni dell'equazione che la loro somma (somma legata alla congettura di Goldbach) permettono di fattorizzare un numero RSA.

Ora un'equazione di secondo grado del tipo:

$$x^2 + ax - b = 0$$

Si può riscrivere come:

$$x(x + a) = b$$

$$x = \frac{b}{a+x} \quad x = -a + \frac{b}{x}$$

In entrambi i casi si può generare una frazione continua:

$$\frac{\frac{b}{a+\frac{b}{a+\frac{b}{a+\dots}}}}{-a+\frac{b}{-a+\frac{b}{-a+\frac{b}{-a+\dots}}}}$$

APPENDICE - ALGORITMI IN PARI/GP

```

/*
* Period of a fraction with
* gaussian function
*
*
*/
GaussN(n)= local(g=0, r=1); {

while( Mod(n,2) == 0, n = n/2;);
while( Mod(n,5) == 0, n = n/5;);

while( g == 0 | r != 1,
      r = Mod(( 10 * r), n);
      g = g + 1;

);

return(g);

}

/*
* IsComposite ?
* -1 Boh
* 1 Yes
* 0 No (it's a prime number)
*
*
*/
isCompN(n)= local(g=0, status=-1); {

if( Mod(n,2) == 0, return(status));
g = GaussN(n);
print("gaussiano(",n,") = ", g);

if( g == n-1,
    status=0;
    print("No, it's a prime number!");
    return(status);
);
if( g != 1,
    if( frac((n-1)/g) != 0.0, status = 1;);
);
};

```

```
    if( status == 1, print("Yes, it's a composite number!"));  
    return(status);  
}  
  
/*  
 * IsMyPrime ?  
 *   1 Yes  
 *   0 No  
 *  
 *  
 */  
isMyPrime(n)= local(status=0); {  
  
    if( eulerphi(n) == n-1, status=1);  
    return(status);  
}
```

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.