Teodoro Cristiano
teodorocristiano@tiscali.it

Un efficiente algoritmo per il calcolo di x^q e di $x^q \pmod p$
(Esponenziazione veloce)

Introduzione

Nel presente articolo viene illustrato un efficiente algoritmo di esponenziazione, noto nella letteratura tecnica con il seguente nome: **right - to - left binary method for exponentiation.**
Di questo algoritmo si danno **due applicazioni** per ciascuna delle quali è stato realizzato un programma in linguaggio Qbasic.
Pertanto come corredo al presente articolo sono disponibili per il lettore anche i due relativi Eseguibili

Prima applicazione

Questa applicazione è relativa al calcolo dell'elevazione a potenza di un numero intero, cioè del calcolo di x^q con i valori sia di ***x*** che di ***q*** numeri interi positivi. Per questo tipo di applicazione si può utilizzare:

- Una *aritmetica a doppia precisione* disponibile nel software adoperato per cui i valori di x e di q non possono essere molto grandi; in effetti, come si può notare dagli esempi riportati più avanti, i valori numerici che superano la soglia di 10^{15} sono dati in virgola mobile e mostrati con mantissa ed esponente. In questa rappresentazione dei numeri vi è quindi un troncamento del risultato numerico alla 16-ma cifra; in compenso viene dato con la parte seguente l'ordine di grandezza del numero.
  Questo però se non viene superato il valore numerico di 2^{1024} pari a $(1.797693134862315) \cdot 10^{308}$, in quanto oltre tale valore si andrebbe in overflow.
- Una *aritmetica a precisione multipla* per cui la notevole limitazione di calcolo utilizzando solo l'aritmetica a doppia precisione può essere superata, L'utilizzo di tale tipo di aritmetica permette di avere risultati di calcolo esatti per qualsiasi valore numerico non eccedente il valore di 10^{14000}, vale a dire per valori numerici aventi fino a 14000 cifre Si possono così gestire ed elaborare con un apposito programma numeri composti anche da diverse centinaia di cifre. Il valore limite 10^{14000} è imposto dalle limitazioni inerenti il software del linguaggio utilizzato (QBasic).

Seconda applicazione

Questa applicazione è dedicata al calcolo di $x^q \pmod p$ con valori interi di ***x, q , p*** sia piccoli che grandi, costituiti ognuno anche da decine o addirittura da qualche centinaio di cifre.

Le due applicazioni indicate, ma soprattutto la seconda, risultano importanti in diversi campi della teoria dei numeri, in particolare nel **campo della Crittografia**.

CALCOLO di x^q

Riguardo alla **prima applicazione** viene illustrato in dettaglio l'algoritmo in questione in quanto esso sarà utilizzato anche nell'altra applicazione. I principali passi che si devono fare riguardo al calcolo della elevazione a potenza di un numero, cioè del calcolo di x^q con valori di x e di q numeri interi positivi, sono elencati qui di seguito:

1) introdurre la variabile x
2) introdurre la variabile q
3) calcolare il valore di q espresso in cifre binarie e memorizzarlo in un vettore;
   si abbia ad esempio il seguente valore binario per q : 110101101 e si indichi con $n = 8$ il numero di cifre binarie che compongono q ; dette cifre verranno memorizzato nel vettore a(j)
   con j = 0, 1, 2, 3, 4, 5, 6, 7, 8 :
   a(8)=1, a(7)=1, a(6)=0, a(5)=1, a(4)=0, a(3) = 1, a(2)=1, a(1)=0, a(0)=1
4) introdurre la variabile c e porre :
   $c = 1$ se q è pari , cioè se a(0) = 0
   $c = x$ se q è dispari, cioè se a(0) = 1
5) formare il seguente ciclo iterativo:
   per ogni valore di j da 1 ad n-1 si effettuino nell'ordine le seguenti operazioni:
   inizio ciclo
   porre $z = x \cdot x$
   se a(j) è di valore 1 porre $y = z \cdot c$ e quindi $c = y$
   porre $x = z$
   tornare all'inizio del ciclo
6) terminato tale ciclo di operazioni l'ultimo valore che si ottiene ottenuto per c è il valore di x^q

Facciamo un semplice esempio:
si voglia trovare il valore di x^{34}
Eseguendo le operazioni indicate nei passi indicati sopra si ha:
q = 34, che espresso in cifre binarie ha il seguente valore: q = 1 0 0 0 1 0
essendo q = 34 numero pari e quindi a(0) = 0 si pone $c = 1$
eseguendo il ciclo per j da 1 sino ad n-1 e visualizzando opportunamente le variabili **j, a(j), z, y, c,** si ha la seguente tabella:

j	a(j)	z	$y = z \cdot c$	$c = y$
1	1	x^2	$x^2 \cdot 1$	x^2
2	0	x^4		
3	0	x^8		
4	0	x^{16}		
5	1	x^{32}	$x^{32} \cdot x^2$	x^{34}

Come si nota l'ultimo valore ottenuto per ***c*** è proprio il valore cercato

Un esempio numerico chiarirà ancora di più l'algoritmo.
Si voglia calcolare il valore di 7^{34}
34 in cifre binarie ha il seguente valore: q = 1 0 0 0 1 0
essendo 34 un numero pari e quindi a(0) = 0 e si pone $c = 1$
eseguendo il ciclo per j da 1 sino ad n-1 e visualizzando opportunamente le variabili **j, a(j), z, y, c** si ha la seguente tabella di valori numerici:

j	a(j)	z	$y = z \cdot c$	$c = y$
1	1	$7 \cdot 7 = 7^2 = 49$	$7^2 \cdot 1 = 49$	$7^2 = 49$
2	0	$7^2 \cdot 7^2 = 7^4 = 2401$		
3	0	$7^4 \cdot 7^4 = 7^8 = 5764801$		
4	0	$7^8 \cdot 7^8 = 7^{16} = 33232930569601$		
5	1	$7^{16} \cdot 7^{16} = 7^{32} = 1.104427674243921 \cdot 10^{27}$	$7^{32} \cdot 7^2 = 7^{34}$	$7^{34} = 5.411695603795212 \cdot 10^{28}$

Come si vede i valori numerici maggiori di 10^{15} sono espressi in virgola mobile e mostrati con mantissa ed esponente.
Diamo qui ora in un linguaggio evoluto, ad esempio in linguaggio Qbasic, un programma relativo al calcolo di x^q relativo alla prima applicazione, limitatamente cioè all'impiego della sola aritmetica a doppia precisione

```
' Programma "X^Q.BAS" (programma con illustrazione didattica)
' Il programma Š relativo a determinare il valore di x^q con l'algoritmo di
' esponenziazione noto come "THE RIGHT TO LEFT BINARY ALGORITHM"
' i valori numeri di x e q non devono essere eccesivamente grandi per
' non eccedere nei risulti il valore massimo di 10^308
CLS : PRINT : PRINT "CALCOLO di  x^q ": DEFDBL A-Z
INPUT "x"; x: INPUT "q"; q
t1 = TIMER
REM  q  viene espresso ora in forma binaria
   ww = LOG(q) / LOG(2): n = INT(ww)+1: PRINT "n="; n
DIM a(n)
   w = q: PRINT " q espresso in binario :";
FOR j = 0 TO n-1: d = INT(w / 2): a(j) = w - 2 * d: w = d: NEXT j
FOR j = n-1 TO 0 STEP -1: PRINT a(j); : NEXT j
REM si effettua ora con le istruzioni sotto riportate l'operazione  x^q
b = x: c = x: IF a(0) = 0 THEN c = 1
PRINT : PRINT "q ="; q, "b ="; b, : PRINT "y ="; c
PRINT : PRINT " j "; "    a(j)", " z "; SPC(26); "y"
s = 0: PRINT
'DO: y$ = INKEY$: LOOP WHILE y$ = ""
REM  inizio ciclo
 FOR j = 1 TO n-1: PRINT j; "    "; a(j),
    z = b * b:  PRINT z,
   IF a(j) = 1 THEN y = c * z: c = y:
   IF a(j) = 1 AND y < 10 ^ 18 THEN s = 14: PRINT SPC(s); y
   IF a(j) = 1 AND y > 10 ^ 18 THEN s = 0: PRINT SPC(s); y
   b = z
   PRINT
NEXT j
Rem fine ciclo
PRINT : PRINT " x ^ q = y ="; y
PRINT : PRINT "CALCOLO DIRETTO DI"; x; "^"; q
PRINT x; "^"; q ; "="; x ^ q
REM fine operazione: e'stato trovato il valore numerico di x^q
PRINT : PRINT "tempo di calcolo ="; TIMER - t1; "secondi"
PRINT "numero di iterazioni:"; j - 1
END
```

In relazione all'utilizzo di detto programma si mostra l'esempio di calcolo di 3^{143} che è un numero di 69 cifre.

Eseguendo il programma si otterrà sullo schermo del monitor il seguente risultato:
(1° esempio)

```
x ? 3
q? 143

q espresso in binario : 1  0  0  0  1  1  1  1
   x = 3      c = x    y = 3

j    a(j)      z                          y = z*c

1    1         9                          27
2    1         81                         2187
3    1         6561                       14348907
4    0         43046721
5    0         1853020188851841
6    0         3.433683820292512D+30
7    1         1.179018457773858D+61      1.691762620188052D+68


y  = 3 ^ 143  = 1.691762620188052D+68
numero di iterazioni : 7
tempo di calcolo = 0 secondi
```

Si ribadisce che i valori numerici che superano la soglia di 10^{15} sono dati in virgola mobile e mostrati con mantissa ed esponente in quanto è stata utilizzata nei calcoli solo la *doppia precisione*.

CALCOLO di x^q con valori di x e q grandi

Per poter calcolare con esattezza x^q con valori grandi di x e q è stato realizzato un ulteriore programma nel quale viene utilizzata un' aritmetica a precisione multipla. Per tale programma, di cui in questo articolo non viene dato il listato, occorre specificare quanto segue:
Se per valori non grandi di x e q le variabili in gioco (vedi x ,q, b, c, z, y) si possono trattare come semplici numeri, ora invece ognuno di questi numeri, che potrebbe essere costituito anche da centinaia di cifre, per poter essere inserito ed elaborato nella sua interezza, deve essere preso in considerazione come stringa alfanumerica da memorizzare poi in un adeguato vettore, nelle cui celle vengono inserite in modo opportuno tutte le cifre del numero stesso. Le operazioni aritmetiche sui diversi vettori saranno poi effettuate con le modalità indicate nell'articolo citato in [CT] .

Date tuttavia anche qui le limitazioni inerenti al software del linguaggio Qbasic utilizzato il valore di x^q non deve superare il valore numerico di 10^{14000}, vale a dire non si possono avere valori numerici costituiti da più di 15000 cifre.
Qui di seguito viene dato un 2° esempio di calcolo con gli stessi valori d i x e q del 1° esempio con il risultato ottenuto utilizzando l'aritmetica a precisione multipla.

[CT] : Cristiano Teodoro – " *sul calcolo della radice quadrata intera di un numero grande*"
Matematicamente.it – Approfondimenti – Matematica

2° ESEMPIO

```
---------------------------------------------- CALCOLO di  x^q ----------------------------------------------
INTRODUCI IL NUMERO  x  E POI BATTI TASTO Invio : 3
INTRODUCI L'ESPONENTE  q  E POI BATTI TASTO Invio : 143
-- ------------------------------------------------------------------------------------------------------------
 3 ^ 143  =  169176 2620188 0520023 9918053 2472321 1896958 0750063 8879624 2120914 7371627

 cifre di 3 ^ 143 : 69
  tempo effettivo di calcolo: 0 secondi
```

nel seguente esempio il calcolo risulta un po' più elaborato ottenendo però tempi di esecuzione molto brevi

3° ESEMPIO : calcolo di 678^{321}

```
---------------------------------------- CALCOLO di x^q ----------------------------------

 INTRODUCI IL NUMERO  x   E POI BATTI TASTO Invio : 678

 INTRODUCI L'ESPONRNTE  q  E POI BATTI TASTO Invio : 321

cifre di  678  ^  321 : 909

-------------------------------------------------------------------------------
se si vuole vedere tutto in una volta il valore completo di 678 ^ 321
battere solo il tasto Invio

se invece si vuole vedere il valore di 678 ^ 321 a gruppi susseguentisi di
di 7 cifre  alla volta battere sulla tastiera 123 e quindi ripetutamente
il tasto Invio
?
                                  678 ^ 321 =

  667931 2013633 5834735 1967706 5132722 1005942 6320872 9249060 3873645 6647022
8066433 4497440 7355283 3600632 3972882 8860641 3198458 4802172 8502332 8428323
1572660 0400377 0205779 5440951 3848515 7340600 0928123 9768998 3952725 8278671
2530697 3148953 7528081 6564568 6931712 4569169 6578522 1673774 7619320 9954509
5299381 5277613 0680027 7276046 8937672 7954996 5843482 1018195 3899143 9920734
5999117 4767728 4308267 6646428 3684245 8366630 8880089 1405023 8127256 7603130
1676307 8068935 3010708 5546369 2623567 8729813 3961062 5747930 7799847 3236697
6337924 0909294 5658753 7813710 2364780 8036471 0309094 3339749 5848567 6183540
4698125 1971539 0939341 1743561 6364290 4644683 6764764 8530179 2724837 6112639
9979607 2124068 0380122 7901478 4349291 8870905 7104917 7606340 1386705 8667818
2548448 6739699 7787081 1632430 5664632 4798737 8259525 4847886 1337619 5124853
5867058 0748604 8996970 6012798 0632060 2007373 5764029 6326375 4124671 9895097
6252351 3421707 2074422 0803779 2505220 1282382 1234770 8938604 1782727 1344128

 cifre di 678 ^ 321 : 909

tempo effettivo di calcolo: .0546875 secondi

SE VUOI UN ALTRO ELENCO DI NUMERI PREMI TASTO s E QUINDI TASTO Invio

SE INVECE VUOI TERMINARE PREMI SOLO TASTO Invio
```

CALCOLO di x^q (mod *p*)

Per quanto riguarda la **seconda applicazione** l'algoritmo per il calcolo di x^q (mod *p*) è il medesimo di quello descritto e impiegato per il calcolo di x^q **.**
La sostanziale differenza risulta essere la seguente:

riguardo le grandezze in gioco e cioè *x, c, z, y* utilizzate nel calcolo di x^q , oltre a tener conto di quanto già specificato, per ogni loro nuovo valore trovato bisogna nei calcoli prendere in considerazione il resto della divisione della grandezza in esame per il numero ***p*** *;* si devono quindi considerare i relativi resti trovati come nuovi valori per *x, c , z* e *y* da utilizzare nella successiva iterazione del loop.

Poiché x^q (mod *p*) viene utilizzato soprattutto per valori numerici relativamente grandi, come avviene ad esempio nel campo della **Crittografia,** risulta importante realizzare un programma che possa manipolare valori di *x,* di *q* e di *p* anche molto elevati ed effettuare su di loro calcoli aritmetici esatti. Pertanto occorre che i valori numerici d'ingresso *x*, *q* e *p* siano opportunamente memorizzati in adeguati arrays (vettori) ed operare su di essi utilizzando una aritmetica a precisione multipla. Si specifica che in questa applicazione tali valori possono essere costituiti ciascuno anche da varie centinaia di cifre.

Per tale applicazione data la importanza di calcolare in modo esatto e veloce il valore di x^q (mod p) in vista delle sue possibili applicazioni nel campo della crittografia si è voluto realizzare e sviluppare un programma ad hoc la cui utilizzazione viene attuata tramite un opportuno Eseguibile.
Con tale programma si possono introdurre e gestire sia per x sia per q sia per p numeri composti ciascuno sino ad un limite massimo di 255 cifre ottenendo per trovare il RESIDUO tempi di calcolo intorno ai 10 secondi. Con leggere modifiche nel listato del programma è possibile andare anche oltre il suddetto limite.
Qui di seguito vengono riportati alcuni esempi di risultati ottenibili col suddetto Eseguibile

4° ESEMPIO

---------- calcolo del residuo di x^q (mod p) ----------

x = 232445364675758645646575759165654566566781 (cifre di x : 42)

q = 10000000000000000000000000000000000000000000000000000000000000000000000000000000000128
(cifre di q : 81)

p = 10000000000000000000000000000000000000000000000000000000000000000000000000000000000129
(cifre di q : 81)

RESIDUO : 1

tempo: .5 secondi

5° ESEMPIO

------------------ --------calcolo del residuo di x^q (mod p) ------------------------

x = 444444444444444444447777766669 (cifre di x : 30)

q = 170141183460469231731687303715884105726 (cifre di q : 39)

p = 170141183460469231731687303715884105727 (cifre di p : 39)

RESIDUO : 1

tempo: .05 secondi

6° ESEMPIO

--- calcolo del residuo di x^q (mod p) ------------------------------------

x = 446456876865645765879756576786743343332323235687807767786656567656566107
(cifre di x : 72)

q = 777777777777777777777777777777777777777777777777777777777777777777777777777777777777
7777777777777777777777777777777777777777800 (cifre di q : 127)

p = 777777777777777777777777777777777777777777777777777777777777777777777777777777777777
7777777777777777777777777777777777777777801 (cifre di p : 127)

RESIDUO : 1
tempo:di calcolo: 3.6796875 secondi

*N.B.: nei 3 esempi sopra riportati il numero **p** è stato scelto primo*

Si vuole mettere in luce con questi 3 ultimi esempi quanto segue :
si valuti con il seguente test la esattezza del risultato trovato:
p è numero primo ; ***q* = *p* -1** ; ***x*** è un numero qualsiasi coprimo rispetto a ***p*.**
Date le suddette condizioni per i valori di ***x*, *q*** e ***p***, *si* può conoscere immediatamente il valore **1** del residuo in quanto vale in questo caso il **Piccolo Teorema** di **Fermat :** $x^{p-1} = 1 \pmod{p}$.

Pertanto riguardo al programma in esame avendo trovato il valore **1** come valore numerico del residuo si può affermare la correttezza dell'algoritmo usato e la esattezza di tutti i calcoli effettuati.

Qui di seguito viene dato come compare sul monitor un ultimo esempio di calcolo dove i valori di x, q e p sono valori scelti in modo casuale:

```
---------------------------------- calcolo del residuo di   x^q (mod p) ----------------------------------------

x = 5465567678764786765465476788909074589934686785345883478456878554345784764546776784545078977798778689665778 66  ( cifre di x : 107 )

q = 566544454687889887067787545754564411124656778905685675556855467567665644454565454656767666767666767  ( cifre di q : 99 )

p = 667454576643544564476845676978553454564433435467896787878976642343233441111122334564565645544565676 7  ( cifre di p : 100 )

 RESIDUO  =  291940064937951343562645492709864745182824485332668572794142168688359012733076882898548800369361589 2   ( cifre  del  RESIDUO:  100 )

tempo di calcolo: 1.26171875 secondi
se vuoi effettuare un altro calcolo scrivi S e quindi premi il tasto Invio
se vuoi uscire devi premere solo il tasto Invio
```