

SOLUZIONE UNIFICATA PER ALCUNE CONGETTURE SUL NUMERO DI PRIMI IN UN CERTO INTERVALLO

Alcune congetture dicono che in un certo intervallo numerico tra due numeri crescenti N ed M (e quindi nell'intervallo numerico $N - M$, anch'esso crescente) e quale che sia la forma aritmetica di N ed M ci sia "almeno" un numero primo.

Esse sono le seguenti:

- **Congettura di Cramer:** "Tra un numero n e $[n + (\ln n)^2]$ esiste sempre un numero primo". In tal caso $M = n$, ed $N = n + (\ln n)^2$
- **Congettura di Oppermann :** " Il numero dei numeri primi fino a $n^2 + n$ è maggiore del numero dei numeri primi fino a n , e questo a sua volta è maggiore del numero dei numeri primi fino a $n^2 - n$ " In tal caso consideriamo per brevità l'intervallo maggiore tra $N = n^2 + n$ ed $M = n^2 - n$.
- **Congettura di Legendre:**" Tra due quadrati perfetti n^2 ed $(n+1)^2$ c'è almeno un numero primo" In tal caso $N = (n+1)^2$ ed $M = n^2$.
- **Congettura o Postulato di Bertrand** (dimostrato da Chebyshev): " Tra un numero $n > 1$ e il suo doppio $2n$ esiste almeno un numero primo" (Versione debole). In tal caso $N = 2n$ ed $M = n$.

Poiché abbiamo trovato una soluzione unica per tutte le suddette congetture, esse sono tutte vere (alcune sono già state dimostrate con altri ragionamenti più complessi).

La nostra soluzione è che poiché l'intervallo $N - M$ cresce con n , più grande è tale intervallo e più numeri primi esso contiene, oltre all'unico numero primo ipotizzato in via generale ("almeno un numero primo").

Considerando i numeri estremi N ed M come sopra definiti per ognuna delle congetture, è chiaro che tra di loro ci sono $k \approx \pi(N) - \pi(M)$ numeri primi, cioè k come differenza crescente tra il numero dei numeri primi fino ad N ed il numero dei numeri primi fino ad M , con N ed M crescenti al crescere del numero n di base.

Poiché tali numeri di numeri primi non si conoscono direttamente, o si contano direttamente sulle tavole dei numeri primi, oppure si stimano con la nota formula di Gauss (o altre più precise), e con essi anche la loro differenza k , che soddisfa tutte le congetture suddette e simili:

$$k \approx \frac{N}{\ln N} - \frac{M}{\ln M} \quad (1)$$

con k approssimato per difetto, per via del termine d'errore del Teorema dei Numeri; e quindi il numero dei numeri primi nell'**intervallo** $I = N - M$ considerato da ciascuna

congettura, è “almeno k numeri primi” per ogni congettura, con k tanto più grande quanto più grande è l’intervallo I.

Come esempio, ne facciamo uno per ognuna delle congettura, con numero di base $N = 100$.

Esempio congettura di Cramer

$$N = 100 + (\ln 100)^2 = 100 + 4,6^2 = 100 + 21,16 = 121,16$$

$$M = 100$$

$$k \approx \frac{121,16}{4,7} - \frac{100}{4,6} = 25,77 - 21,73 = 4,04$$

In realtà, con i valori reali, abbiamo $k = 30 - 25 = 5 \approx 4,04$ con cinque numeri primi (101, 103, 107, 109, e 113) tra 100 e 121 (tralasciando i decimali 0,16). Errore k reale - k calcolato = $5 - 4 = 1$, cioè di un solo numero primo su cinque reali.

Esempio congettura di Oppermann

$$n = 10; \quad N = 10^2 + 10 = 100 + 10 = 110$$

$$M = 10^2 - 10 = 100 - 10 = 90$$

$$k \approx \frac{110}{\ln(110)} - \frac{90}{\ln(90)} = \frac{110}{4,7} - \frac{90}{4,9} = 23,40 - 20,04 = 3,3$$

in realtà abbiamo $5 \approx 3,3$ numeri primi (97, 101, 103, 107, 109) tra 90 e 110, dei quali solo uno (97) tra 90 e 100, e ben quattro tra 100 e 110 : (101, 103, 107, 109), che soddisfano la congettura (fino a 90 ci sono 24 numeri primi, fino a 100 ce ne sono 25, fino a 110 ce ne sono 29, con 24, 25 e 29 crescenti con 90, 100 e 110).

Esempio congettura di Legendre

$$M = 10^2 = 100; \quad N = (10 + 1)^2 = 121 = \text{quadrato successivo}$$

$$k \approx \frac{121}{\ln(121)} - \frac{100}{\ln(100)} = \frac{121}{4,79} - \frac{100}{4,60} = 25,26 - 21,7 = 3,56$$

In realtà tra 100 e 121 ci sono i $5 \approx 3,56$ numeri primi 101, 103, 107, 109, 113, che soddisfano la congettura, essendo più di uno (congettura soddisfatta anche per tutti gli altri n minori e maggiori di 10).

Esempio congettura o postulato di Bertrand

$$M = 100; \quad N = 2M = 200.$$

$$k \approx \frac{200}{\ln(200)} - \frac{100}{\ln(100)} = \frac{200}{5,29} - \frac{100}{4,60} = 37,80 - 21,73 = 16,07$$

In realtà tra 100 e 200 ci sono ben $21 \approx 16,07$ numeri primi (essendo $200 - 100 = 100$ l’intervallo più lungo rispetto alle congetture precedenti, e quindi più ricco di numeri

primi) Anche qui, la congettura di Bertrand è soddisfatta, e anche per tutti gli altri $n > 1$, come si può facilmente verificare con la suddetta formula unica per tutte le congetture che si basano su un intervallo $I = N - M$, quale che sia la forma aritmetica di N ed M rispetto ad un numero minore n .

La conclusione è che tutte le suddette congetture sono vere, e verificabili con la formula unica per almeno k numeri primi (con k stimato con la suddetta formula) nell'intervallo I tipico per ciascuna congettura, salve le precedenti dimostrazioni per ognuna di esse e basate su altri ragionamenti matematici. E se, infine, si volessero correggere i valori di k così

stimati con il numero correttore $c' = 1,15048$ per $N = 10^2 = 100$ (Vedi il nostro precedente lavoro "Due formule più precise per il calcolo dell' N -esimo numero primo e per $\pi(N)$ ", già su questo sito), si otterrebbero rispettivamente valori più vicini ai valori reali di k , infatti:

k stim. x c'	k reale	congettura
4,04 x 1,15048 = 4,64	5	Cramer
3,3 x 1,15048 = 3,79	5	Oppermann
3,56 x 1,15048 = 4,095	5	Legendre
16,07 x 1,15048 = 18,48	21	Bertrand

Risultati leggermente migliori si ottengono invece correggendo con $c' = 1,15048$, anziché la differenza finale, i singoli termini delle rispettive differenze, ottenendo ora rispettivamente le differenze finali seguenti per le stime di k , comunque sempre per difetto: 4,65 per Cramer; 3,87 per Oppermann; 4,1 per Legendre; 18,49 per Bertrand, con rispettivi valori reali di k : 5,5, 5,21.

Concludendo, le suddette quattro congetture, essendo vere e dimostrate possono essere ora considerate teoremi veri e propri: "Tra i numeri N ed M previsti da ciascuna delle ex-congetture, ci sono almeno k primi, con k reale $k = \pi(N) - \pi(M)$ e k stimato con la formula (1), che dà risultati approssimati per eccesso, anche se leggermente corretta con il numero correttore c' ".

**Prof. Annarita Tulumello del
GRUPPO ERATOSTENE**

APPENDICE SOFTWARE

ing. R. Turco

L'algoritmo presentato calcola il numero di primi presenti nell'intervallo definito dalle congetture e teoremi di Cramer, Oppermann, Legendre e Bertrand. Inoltre si calcola il valore proposto dal gruppo ERATOSTENE senza e con correttore.

E' innanzitutto un algoritmo didattico che utilizza interi "unsigned long int", per cui l'algoritmo nella rappresentazione degli interi dipende dal limite architetturale del hardware disponibile: 32 bit o 64 bit. Nel caso si voglia superare tale limite hardware è necessario utilizzare librerie software che rappresentano tipi verylong (Vedi articolo "Analisi computazionale – Metodo di superamento dei limiti hardware a 32 e 64 bit" http://www.geocities.com/SiliconValley/Port/3264/corsi/matematica/ANC_OMP.pdf). Ad esempio utilizzando un hardware a 32 bit si possono rappresentare, senza causare overflow, interi solo fino a $2^{32}-1$.

Nel seguito viene presentato l'algoritmo. La sua efficienza è basata soprattutto nel ricavare in un solo passaggio il numero di primi tra i due valori forniti dalle congetture, per evitare di ricontare due volte i primi comuni ai due valori dell'intervallo. Maggiore è il numero introdotto e maggiore è il tempo necessario da attendere, soprattutto a causa dei quadrati introdotti dalle congetture di Oppermann e Legendre.

Sorgente main.c

```
#include <stdio.h>
#include <stdlib.h>

//
// R. Turco (C) 2008
// Calcolo Delta P(N): Cramer, Oppermann, Legendre, Bertrand, Eratostene
// Main
//

// prototype
unsigned long int DiffCountPidiN(unsigned long int min, unsigned long int sup);

int main(int argc, char *argv[])
{
    unsigned long int Num1=0, Num2=0;
    unsigned long int pdn1=0, pdn2=0;
    double pot=0.0,pot1=0.0, era=0.0, era1=0.0, cor=0;
    int scelta=0;

    unsigned long int vbert=0, vcram=0, vcram1=0, vopper=0, vlegendre=0;

    printf("\nCalcolo delta Pi(N) - R. Turco (C) 2008");
    printf("\nLimite di rappresentazione interi 2^32-1");

    do{

        printf("\n\nScegli congettura (0 per uscire): \n");
        printf("\n1. Cramer");
        printf("\n2. Oppermann");
        printf("\n3. Legendre");
        printf("\n4. Bertrand\n\n");
        scanf("%d",&scelta);
```

```

if( scelta <= 0 ) return; // Si esce
if( scelta > 4 ) return;

printf("\nInserisci numero di inizio (diverso da 0) : \n\n");
scanf("%d",&Num1);

if( Num1 <= 0 ) return; // Si esce

if( Num1 <= 10 ) cor=0.9216;
if( (Num1 > 10) && (Num1 <= 100)) cor=1.1504;
if( (Num1 > 100) && (Num1 <= 1000)) cor=1.1592;
if( (Num1 > 1000) && (Num1 <= 10000)) cor=1.1319;
if( (Num1 > 10000) && (Num1 <= 100000)) cor=1.1040;
if( (Num1 > 100000) && (Num1 <= 1000000)) cor=1.0844;

printf("\ncorrettore c' = %f\n", cor);
switch(scelta){

case 1:

// Cramer x, x+ln(x)^2
printf("\nCramer: x, x+ln(x)^2\n");
pot = pow(log((double)Num1),2);
vcram1 = Num1 + pot;
era = ((Num1 + pot)/log(Num1+pot)) - (Num1/log(Num1));
vcram = DiffCountPidiN(Num1, vcram1);
printf("Cramer: Pi(%d) - Pi(%d) = %d\n", vcram1, Num1, vcram);
printf("formula Eratostene: sup/ln(sup) - inf/ln(inf) = %f\n", era);
printf("valore corretto con c' => %f\n", era*cor);

break;

case 2:

// Oppermann x^2 - x, x^2 + x
pot=0.0;
printf("\nOpperman: x^2-x , x^2+x\n");
pot=pow((double)Num1,2);
pdn1=pot+Num1;
pdn2=pot-Num1;
era=((pot+Num1)/log(pot+Num1));
era1=((pot-Num1)/log(pot-Num1));
vopper = DiffCountPidiN(pdn2,pdn1);
printf("Oppermann: Pi(%d) - Pi(%d) = %d\n", pdn1, pdn2, vopper);
printf("formula Eratostene: sup/ln(sup) - inf/ln(inf) = %f\n", era - era1);
printf("valore corretto con c' => %f\n", (era-era1)*cor);

break;

case 3:

// Legendre x^2, (x+1)^2
pot=0.0;
pot1=0.0;
pot =pow((double)Num1,2);
pot1=pow((double)(Num1+1),2);
pdn1=pot;
pdn2=pot1;
era=(pot1/log(pot1)) - (pot/log(pot));
printf("\nLegendre: x^2, (x+1)^2\n");

```

```

vlegendre = DiffCountPidiN(pdn1,pdn2);
printf("Legendre: Pi(%d) - Pi(%d) = %d\n", pdn1, pdn2, vlegendre);
printf("formula Eratostene: sup/ln(sup) - inf/ln(inf) = %f\n", era);
printf("valore corretto con c' => %f\n", era*cor);

break;

case 4:
// Bertrand x, 2x
printf("\nBertrand: x, 2x\n");
vbert = DiffCountPidiN(Num1,2*Num1);
printf("Bertrand: Pi(%d) - Pi(%d) = %d\n", 2*Num1, Num1, vbert);
era=((2*Num1)/log((double)(2*Num1)));
era1= (Num1/log((double)Num1));
printf("formula Eratostene: sup/ln(sup) - inf/ln(inf) = %f\n", era-era1 ); // Eratostene
printf("valore corretto con c' => %f\n", (era-era1)*cor);
break;

default:
break;
}

system("PAUSE");

}while(Num1 != 0 );

return 0;
}

```

Sorgente diffpidin.c

```

#include "prototype.h"

unsigned long int DiffCountPidiN(unsigned long int min, unsigned long int sup){

/* Conta il numero di primi tra min e sup a partire da 2*/
/* iMin conta i numeri primi trovati fino a min */

unsigned long int iCounter=0, iMin=0, i=0;
for(i=2; i<sup; i++){
if( prime(i,0) != 0 ){
iCounter++;
}
if( i == min ){
iMin = iCounter; /* trovato il #primi per min */
}
}
/* La differenza da il numero di primi tra min e sup */
return (iCounter - iMin);
}

```

Sorgente prime.c

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

unsigned long int prime(unsigned long int iVal, int iDeb)
{
int i=0, iCount=0;
unsigned long int iSq=0, iPri=0;

/* iCount conta i Divisori */

```

```

float fVal = iVal;

/* Se 0,1,2,3 esco rapidamente */

switch(iVal){
  case 0:
  case 1:
    return iPri;
  break;
  case 2:
    return 1;
  case 3:
    return 2;
  break;
  default:
  break;
}

iSq = sqrt(fVal);

if( iDeb > 0 ){
  printf("\n sqrt : %d\n", iSq);
}

if( iVal%2 == 0 ){
  iCount++;
}
else {
  /* Cerco i divisori con il modulo escludendo l'1 e il 2*/
  for(i=3;i<=iSq;i=i+2){ /* Cerco i divisori dispari */
    if( iVal%i == 0 ) {
      iCount++; /* Esiste un divisore, allora non è primo */
      break; /* Interrompo al primo Divisore trovato */
    }
  }
}
if( iCount == 0 ) iPri=1; /* Se iCount = 0 allora è primo */

return iPri;
}

```

Sorgente prototype.h

```

#ifndef prototype_prime
#define prototype_prime

static unsigned long int NumItem=0;
typedef struct{
  unsigned long int PiInit1;
  unsigned long int PiEnd1;
  unsigned long int PiInit2;
  unsigned long int PiEnd2;
}SUBSET;

int prime(unsigned long int iVal, int iDeb);
unsigned long int DiffCountPidiN(unsigned long int min, unsigned long int sup);

unsigned long int CountPidiN(unsigned long int iValue);
int SxDuePrimiOpt(unsigned long int iValue, int smp);
void SxDuePrimiSubset(SUBSET *p, unsigned long int iValue);
unsigned long int gen6n(unsigned long int Value);

```

```
unsigned long int gen2n(unsigned long int Value);  
void genprime(unsigned long int Pos);
```

```
#endif
```