

Riconoscimento di numeri primi

Marcello Colozzo - <http://www.extrabyte.info>

Per il riconoscimento del tipo di numero (**Integer**, **Real**, **Complex**, **Prime**), *Mathematica* utilizza un comando del tipo ***Q[]**, dove al posto dell'asterisco va messo **Integer**, etc. Quindi, per i numeri primi il comando è **PrimeQ[]**. Ad esempio:

```
In[1]:= PrimeQ[11 003]
```

```
Out[1]= True
```

```
In[2]:= PrimeQ[11 009]
```

```
Out[2]= False
```

Per il riconoscimento di numeri primi, *Mathematica* utilizza il cosiddetto test forte di 2-pseudoprimi. Quest'ultimo, però, non è un certificato di primalità affidabile, per cui *Mathematica* utilizza anche un test basato sugli pseudoprimi di Lucas (almeno nelle "vecchie" versioni). Stan Wagon nel suo libro *Guida a Mathematica* propone un Package che velocizza il certificato di primalità utilizzato da *Mathematica*, attraverso la definizione del comando **primeQ[]** (si ricordi che *Mathematica* è case sensitive). Il Package è per la vecchia versione 2.0. Noi l'abbiamo modificato in modo da renderla compatibile alla 6.0 e superiori. Ecco il package.

```
In[3]:= BeginPackage["primeQ`"];
```

```
In[4]:= smallprimes = Prime[Range[1000]];
```

```
In[5]:= testforsmalldivisors[x_] :=  
  (n = 1; While[n < 1000 && Mod[x, smallprimes[[n]]] ≠ 0, n++]; n == 1000)
```

```
In[6]:= primeQ[n_] := (n ≤ 7919 || testforsmalldivisors[n] && PrimeQ[n]);
```

```
In[7]:= EndPackage[];
```

Utilizzando il comando **Timing[]** è possibile confrontare i tempi di computazione. A tale scopo carichiamo il package:

```
In[8]:= Needs["primeQ`"]
```

Quindi, applichiamo il nuovo comando a $10^{200} + 5$

```
In[9]:= primeQ[10200 + 5] // Timing
```

```
Out[9]= {0., False}
```

```
In[10]:= PrimeQ[10200 + 5] // Timing
```

```
Out[10]= {0.031, False}
```

da cui vediamo che **primeQ** è molto più veloce di **PrimeQ**.