

Esercizio guidato

Settimana 5
03/11/2022

Si ringrazia il Dott. Giacomo Baruzzo per il materiale

Il crivello di Eratostene

Il *crivello di Eratostene* è un noto algoritmo per la ricerca dei numeri primi **minori di un certo valore massimo MAX**

L'algoritmo prevede i seguenti passi:

- si considerano tutti i numeri naturali a partire da 2 fino **MAX**
- per ogni numero *i* da 2 a $\text{MAX} - 1$, si cancellano tutti i multipli di *i*
- i numeri che restano sono i numeri primi minori di MAX

Il crivello di Eratostene

	2	3	4	5	6	7	8	9	10	Prime numbers
11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	
31	32	33	34	35	36	37	38	39	40	
41	42	43	44	45	46	47	48	49	50	
51	52	53	54	55	56	57	58	59	60	
61	62	63	64	65	66	67	68	69	70	
71	72	73	74	75	76	77	78	79	80	
81	82	83	84	85	86	87	88	89	90	
91	92	93	94	95	96	97	98	99	100	
101	102	103	104	105	106	107	108	109	110	
111	112	113	114	115	116	117	118	119	120	

Esempio

MAX = 12

Lista: 2,3,4,5,6,7,8,9,10,11 ← candidati numeri primi minori di MAX

- Iterazione 1: elimino i multipli di “2”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 2: elimino i multipli di “3”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 3: elimino i multipli di “4”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 4: elimino i multipli di “5”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 5: elimino i multipli di “6”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 6: elimino i multipli di “7”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 7: elimino i multipli di “8”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 8: elimino i multipli di “9”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 9: elimino i multipli di “10”: 2,3,4,5,6,7,8,9,10,11
- Iterazione 10: elimino i multipli di “11”: 2,3,4,5,6,7,8,9,10,11

I numeri che restano sono i numeri primi minori di MAX: 2, 3, 5, 7, 11

Idea di soluzione: usiamo le liste

- si predispone una lista di **MAX** valori *booleani*
- ogni elemento della lista "rappresenta" il numero intero corrispondente al proprio indice nella lista: se e solo se l'elemento è **False**, allora il numero corrispondente è stato ELIMINATO dall'insieme dei numeri primi, cioè non è un numero primo
- all'inizio si suppone che tutti i numeri siano primi
- successivamente, si considera ciascun numero intero *i* maggiore di uno, in ordine crescente, e si "eliminano" tutti i numeri che ne sono multipli, contrassegnandoli con **False** nella lista (ad esempio, al primo passaggio vengono "eliminati" tutti i numeri pari; al secondo passaggio vengono "eliminati" tutti i multipli di 3, e così via)
- al termine, i numeri rimasti (contrassegnati da **True**) sono tutti e soli i numeri primi cercati, non essendo multipli di alcun numero

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Chiedo MAX come input e controllo sia un intero positivo

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Chiedo MAX come input e controllo sia un intero positivo

Verifichiamo cosa succede se inserisco in input:

- una stringa
- un numero decimale (float)
- un numero negativo
- un numero intero positivo

Dall'algoritmo al codice

```
from myinput import *
MAX = inputPositiveDecimalInteger("Un
numero intero positivo: ")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Importo il modulo
myinput (lab successivo)
Chiedo MAX come input
usando la nostra funzione

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Creo una lista di MAX
elementi inizializzati a True

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Considero i numeri *i* da 2 a MAX-1

Infatti, gli indici (valori) 0 e 1 non sono interessanti!

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Contrassegno con False i
multipli di *i*

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Stampo i numeri primi

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Verificare il corretto funzionamento del programma con:

- MAX = 1 e MAX = 2 (non viene visualizzato nessun numero, perché non esiste nessun numero primo minore di 1 o 2)
- MAX = 3 (viene visualizzato soltanto il numero 2)
- MAX = 4 e MAX = 5 (vengono visualizzati soltanto i numeri 2 e 3)
- MAX = 6 (vengono visualizzati soltanto i numeri 2, 3 e 5)

Dall'algoritmo al codice

```
from sys import exit
MAX = int(input("Un numero intero
positivo: "))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Possiamo osservare che le prime due celle della lista, ovvero quelle corrispondenti ai numeri 0 e 1, sono di fatto sempre "sprecate".

per risparmiare, potrei usare MAX-2 come dimensione della lista, e poi dire che il numero x è rappresentato dalla cella di indice x-2, ma per due sole celle non vale la pena

Soluzione 1

```
from sys import exit
MAX = int(input("Un numero intero positivo:
"))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Soluzione 2

```
from sys import exit
MAX = int(input("Un numero intero positivo:
"))
if MAX <= 0 :
    exit("Ho detto positivo!")
primes = [True] * MAX
for i in range(2, MAX//2) :
    for j in range(2*i, MAX, i) :
        primes[j] = False
print("Numeri primi minori di %i:" % MAX)
for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')
print()
```

Nel primo ciclo, il valore finale di i potrebbe essere limitato a $MAX/2$: numeri maggiori o uguali a $MAX/2$ hanno il loro primo multiplo (il doppio) che è già maggiore di $MAX-1$, quindi non "eliminano" nessun altro numero

Analizziamo meglio l'algoritmo

MAX = 11

Lista: 2,3,4,5,6,7,8,9,10

- Iterazione 1: elimino i multipli di "2": **2,3,4,5,6,7,8,9,10**
- Iterazione 2: elimino i multipli di "3": **2,3,4,5,6,7,8,9,10**
- Iterazione 3: elimino i multipli di "4": **2,3,4,5,6,7,8,9,10**
- Iterazione 4: elimino i multipli di "5": **2,3,4,5,6,7,8,9,10**

Analizziamo meglio l'algoritmo

MAX = 11

Lista: 2,3,4,5,6,7,8,9,10

- Iterazione 1: elimino i multipli di "2": **2,3,4,5,6,7,8,9,10**
- Iterazione 2: elimino i multipli di "3": **2,3,4,5,6,7,8,9,10**
- Iterazione 3: elimino i multipli di "4": **2,3,4,5,6,7,8,9,10**
- Iterazione 4: elimino i multipli di "5": **2,3,4,5,6,7,8,9,10**

Dobbiamo veramente analizzare i multipli di 4?

No, perché tutti i multipli di 4 sono anche multipli di 2, quindi sono già stati eliminati alla iterazione 1!

Analizziamo meglio l'algoritmo

MAX = 11

Lista: 2,3,4,5,6,7,8,9,10

- Iterazione 1: elimino i multipli di "2": **2,3,4,5,6,7,8,9,10**
- Iterazione 2: elimino i multipli di "3": **2,3,4,5,6,7,8,9,10**
- Iterazione 3: elimino i multipli di "4": **2,3,4,5,6,7,8,9,10**
- Iterazione 4: elimino i multipli di "5": **2,3,4,5,6,7,8,9,10**

Dobbiamo veramente analizzare i multipli di 4?

No, perché tutti i multipli di 4 sono anche multipli di 2, quindi sono già stati eliminati alla iterazione 1!

IDEA: possiamo quindi evitare di analizzare i multipli di un numero, se questo stesso numero è già stato eliminato in una iterazione precedente¹⁸

Soluzione 2

```
from sys import exit

MAX = int(input("Un numero intero positivo: "))

if MAX <= 0 :
    exit("Ho detto positivo!")

primes = [True] * MAX

for i in range(2, MAX//2) :
    for j in range(2*i, MAX, i) :
        primes[j] = False

print("Numeri primi minori di %i:" % MAX)

for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')

print()
```

Soluzione 3

```
from sys import exit

MAX = int(input("Un numero intero positivo: "))

if MAX <= 0 :
    exit("Ho detto positivo!")

primes = [True] * MAX

for i in range(2, MAX//2) :
    if primes[i] :
        for j in range(2*i, MAX, i) :
            primes[j] = False

print("Numeri primi minori di %i:" % MAX)

for i in range(2, MAX) :
    if primes[i] :
        print(i, end=' ')

print()
```

Con MAX = 100, il nuovo algoritmo evita di eliminare “inutilmente” tutti i multipli di: 4 6 8 9 10 12 14 15 16 18 20 21 22 24 25 26 27 28 30 32 33 34 35 36 38 39 40 42 44 45 46 48 49