

Sieve of Eratosthenes

The *Sieve of Eratosthenes* is an algorithm that dates back to about 240 BC. It can be used to list primes and is one of the oldest methods of its kind. This elegant method is particularly effective when used for smaller primes. However, the method requires keeping a list of all numbers smaller than or equal to a predetermined upper limit n . This is an increasing disadvantage as the primes to be determined get larger.

| 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 |

Demonstration of the method used in the Sieve of Eratosthenes for finding prime numbers smaller than or equal to 120.

The method to find the list of primes that are smaller or equal to a predefined integer n works as follows:

1. Make a list of all the numbers from 2 to n .
2. Define m as the smallest number from the list.
3. Cross out all multiples of m in the list, without crossing out m itself.
4. Define m as the next number from the list.
5. Continue from step 3, or stop if no next number m can be determined.

The numbers which were eventually not crossed off are all prime numbers smaller than or equal to n . This procedure can be sped up in several ways:

- In step 4 it is useless to choose a number that has already been crossed off, because all its multiples are already crossed off in the previous steps.
- You can start with crossing of the multiples of m from m^2 . All smaller multiples are already crossed off in the previous steps.
- If m^2 is larger than n , than the procedure can be stopped.

Let us clarify the procedure above with an example, in which we seek the prime numbers smaller or equal to $n = 30$. We start with a list of 2 until and including 30.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

In the first round we cross off the multiples of $m = 2$, starting with $2^2 = 4$.

~~2~~ 3 ~~4~~ 5 ~~6~~ 7 8 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27 ~~28~~ 29 30

Then we continue crossing of the multiples of $m = 3$, starting with $3^2 = 9$. Because some numbers can be crossed off multiple times we will now display the multiples of m that are

crossed off in the next step in green. The numbers displayed in grey are multiples of m that are smaller than m^2 . These numbers can be skipped, because they have already been crossed off in the previous steps.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The number $m = 4$ has already been crossed off. We continue crossing off the multiples of $m = 5$, starting with $5^2 = 25$.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The next number that has not been crossed off yet is $m = 7$. We should now cross off the multiples of 7, starting with 7^2 . This square, however, is larger than 30, which means the procedure stops here. The result is the prime numbers smaller or equal to $n = 30$.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Assignment

Write a function `sieve` to which a number $n \in \mathbb{N}$ should be passed as argument. This function must return a sorted list that contains all prime numbers smaller or equal to n . For this the function the Sieve of Eratosthenes should be implemented, including the optimization to speed up the procedure of crossing off.

Example

```
>>> sieve(6)
[2, 3, 5]
>>> sieve(30)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
>>> sieve(100)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

De *Zeef van Eratosthenes* is een algoritme dat dateert van circa 240 voor Christus. Het kan gebruikt worden om priemgetallen op te lijsten en is daarmee één van de oudste methoden van zijn soort. Deze elegante methode is vooral efficiënt wanneer ze wordt gebruikt voor kleinere priemgetallen. De methode vergt echter het bijhouden van een lijst met alle getallen die kleiner zijn of gelijk aan een vooraf vastgelegde bovengrens n . Dit wordt een steeds groter nadeel naarmate de te bepalen priemgetallen groter worden.

| | 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 | |

Demonstratie van de methode die gebruikt wordt bij de Zeef van Eratosthenes voor het vinden van de priemgetallen die kleiner of gelijk zijn aan 120.

De methode om de lijst van priemgetallen te vinden die kleiner zijn of gelijk aan een vooraf opgegeven getal n werkt als volgt:

1. Maak een lijst van alle getallen van 2 tot en met n .
2. Bepaal m als het kleinste getal uit de lijst.
3. Doorstreep in de lijst alle veelvouden van m , zonder daarbij m zelf te doorstrepen.
4. Bepaal m als het volgende getal uit de lijst.
5. Ga verder vanaf stap 3, of stop indien er geen volgende getal m meer kan bepaald worden.

De getallen die uiteindelijk niet doorstreept werden, zijn alle priemgetallen die kleiner zijn of gelijk aan n . Deze procedure kan nog op enkele manieren versneld worden:

- Het heeft geen zin om in stap 4 een getal te kiezen dat al doorstreept is, want alle veelvouden daarvan zijn reeds doorstreept in voorgaande stappen.
- Men kan met het doorstrepen van de veelvouden van m beginnen vanaf m^2 . Alle kleinere veelvouden zijn reeds doorstreept in voorgaande stappen.
- Als m^2 groter is dan n , dan kan de procedure gestopt worden.

We verduidelijken bovenstaande procedure met een voorbeeld, waarbij we de priemgetallen zoeken die kleiner zijn of gelijk aan $n = 30$. Daarvoor beginnen we met een lijst van 2 tot en met 30.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

In de eerste ronde strepen we de veelvouden van $m = 2$ weg, te beginnen met $2^2 = 4$.

~~2~~ 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27 ~~28~~ 29 ~~30~~

Daarna gaan we verder met het schappen van de veelvouden van $m = 3$, te beginnen bij $3^2 = 9$. Omdat sommige getallen meerdere keren kunnen geschrappt worden, zullen we in wat volgt de veelvouden van m die in de volgende stap geschrappt worden in het groen weergeven. De getallen die in het grijs weergegeven worden, zijn veelvouden van m die kleiner zijn dan m^2 . Deze getallen kunnen overgeslaan worden, omdat ze reeds doorstreept werden in voorgaande stappen.

2 3 4 5 ~~6~~ 7 8 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 16 17 ~~18~~ 19 20 ~~21~~ 22 23 ~~24~~ 25 26 ~~27~~ 28 29 ~~30~~

Het getal $m = 4$ is al weggestreept. We gaan dus verder met het schappen van de veelvouden van $m = 5$, te beginnen bij $5^2 = 25$.

2 3 4 ~~5~~ 6 7 8 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 16 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 24 ~~25~~ 26 27 ~~28~~ 29 ~~30~~

Het volgende getal dat nog niet geschrappt werd, is $m = 7$. We zouden nu dus de veelvouden van 7 moeten wegstrepen, te beginnen met 7^2 . Dit kwadraat is echter groter dan 30 en dus kunnen we de procedure hier stoppen. Wat we overhouden zijn de priemgetallen die kleiner zijn of gelijk aan $n = 30$.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Opgave

Schrijf een functie `zeef` waaraan een getal $n \in \mathbb{N}$ als argument moet doorgegeven worden. Deze functie moet een gesorteerde lijst teruggeven die alle priemgetallen bevat die kleiner zijn of gelijk aan n . Hiervoor moet de functie de Zeef van Eratosthenes implementeren, inclusief de optimalisatie om de procedure van het schrappen te versnellen.

Voorbeeld

```
>>> zeef(6)
[2, 3, 5]
>>> zeef(30)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
>>> zeef(100)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```