

Voorbeeld van het gebruik van lijsten: Zeef van Eratosthenes

Priemgetallen werden reeds door de Grieken bestudeerd. Zij ontwikkelden een methode om priemgetallen op te lijsten die efficiënter is dan volgend algoritme:

In [1]:

```
# Kijk na of een getal x een priemgetal is
x=int(input("Geef een getal:"))
if x<2:
    print("Geen priemgetal want kleiner dan 2") # 0 en 1 zijn geen priemgetallen
delerGevonden=False
for d in range(2,x):
    if x%d==0: # als d een deler is van x ...
        print("Geen priemgetal; ",d,"is een deler") # ... is het geen priemgetal
        delerGevonden=True
if not delerGevonden:
    print(x,"is een priemgetal!")
```

13 is een priemgetal!

De Griekse methode die we gaan implementeren heet *de zeef van Eratosthenes* en werkt als volgt. Om alle priemgetallen van 1 tot N te berekenen:

- schrijf alle getallen van 2 tot N neer in een tabel
- ga alle getallen in het rooster een voor een af
 - als het getal niet doorgestreept is:
 - je hebt een priemgetal gevonden
 - streep alle veelvouden van dit getal door

Als we bijvoorbeeld alle priemgetallen kleiner dan 10 willen, schrijven we alle getallen van 2 tot 9 op:

2 3 4 5 6 7 8 9

Vervolgens starten we met het kleinste getal (2); dit is een priemgetal. We strepen alle getallen door die een veelvoud zijn van dat getal:

2 3 / 5 / 7 / 9

Dan gaan we verder tot we het volgende niet doorstreepte getal tegenkomen (3). Dit is opnieuw een priemgetal. We strepen opnieuw alle veelvouden door:

2 3 / 5 / 7 / /

enzovoort ... Volgende illustratie geeft deze procedure weer voor alle getallen tot 120.

	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	

CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=2810935>
 (<https://commons.wikimedia.org/w/index.php?curid=2810935>)

We gaan deze zeef nu zelf implementeren in Python. Onze eerste keuze is welke *datastructuur* we gaan gebruiken. Het ligt voor de hand om een lijst te gebruiken om de zeef voor te stellen. We zouden in deze lijst alle getallen van 2 tot N kunnen stoppen:

In []:

```
N=120 #aantal elementen in de zeef
Zeef=[]
for i in range(2,N+1):
    Zeef.append(i)
print(Zeef)
```

Vervolgens moeten we nadenken hoe we kunnen opslaan of een getal doorstreept is of niet. We zouden een doorstreept getal kunnen vervangen door -1. Bijvoorbeeld, alle veelvouden van 2 doorstrepen kan dan bijvoorbeeld als volgt:

In []:

```
for i in range(3,N+1): # We willen 2 zelf niet doorstrepen, dus starten bij 3
    if i%2==0: # rest bij deling door 2 = 0
        # i staat op plaats i-2 in de zeef
        Zeef[i-2]=-1

print(Zeef)
```

We kunnen ook rechtstreeks alle veelvouden van 2 doorstrepen; dus zonder de hele zeef af te lopen en telkens te testen of het een veelvoud van 2 is:

In []:

```
N=120 #aantal elementen in de zeef
Zeef=[]
for i in range(2,N+1):
    Zeef.append(i)

for i in range(2*2,N+1,2):
    Zeef[i-2]=-1

print(Zeef)
```

Of, algemener, alle veelvouden van i schrappen:

In []:

```
i=3
# Schrap alle veelvouden van i uit de zeef Z die van 2 tot N gaat
for j in range(2*i,N+1,i):
    Zeef[j-2]=-1

print(Zeef)
```

Dat schrappen moeten we nu doen voor elk getal in de zeef, van klein naar groot, dat nog niet zelf geschrapt is:

In []:

```
N=120 #aantal elementen in de zeef
Zeef=[]
for i in range(2,N+1):
    Zeef.append(i)

for getal in Zeef:
    if getal!=-1:
        print(getal,"is een priemgetal!")
        for j in range(2*getal,N+1,getal):
            Zeef[j-2]=-1
```

Hiermee is ons programma af.

Alternatief

Afhankelijk van welke datastructuur we gebruikten hadden we een variant gekregen van hetzelfde algoritme. Bijvoorbeeld, we konden ook volgende datastructuur gebruiken: een lijst met Booleaanse waarden (*True* of *False*), waarbij *Zeef[i]==True* betekent: *i* is doorstreept. we beginnen met de lijst [*False, False, True, True, ...*], want 0 en 1 zijn sowieso geen priemgetallen. Vervolgens gaan we over alle getallen van 2 tot N en strepen de veelvouden door van niet-geschrapte getallen. Op het einde zijn alle getallen *i* zodat *Zeef[i]==False*, priemgetallen:

In []:

```
N=120
Zeef=[True,True]+[False]*(N-1) # Net zoals bij strings kunnen we met * een lijst een a
antal keer achter elkaar kopiëren

for i in range(2,N+1):
    if not Zeef[i]: # niet doorstreept
        print(i,"is priem")
        for j in range(2*i,N+1,i): # Loop alle veelvouden van i af
            Zeef[j]=True # doorstreep het veelvoud
```